**Федеральное агентство связи**

**Государственное бюджетное образовательное учреждение высшего профессионального образования**

**ордена Трудового Красного Знамени**

**Московский технический университет связи и информатики**

**Кафедра информатики**

**Курсовая работа по дисциплине “Основы алгоритмизации и программирования”**

**По теме «Использование электронного пособия в учебном процессе на примере дисциплины “Английский язык”»**

Выполнили студенты группы БФИ1802

Копиевский Н. Ю.

Сиротин Н. С.

Научный руководитель:

доцент кафедры Информатика

к.п.н. Гуриков С. Р.

доцент кафедры Иностранные языки

к.п.н Кожевникова Т.В.

**Москва 2019**

**Оглавление**

[**ВВЕДЕНИЕ** 2](#_Toc10301547)

[**Глава I. Постановка задачи** 4](#_Toc10301548)

[**1.1** **Техническое задание на разработку программного продукта** 4](#_Toc10301549)

[**1.1.1 Основания для разработки** 4](#_Toc10301550)

[**1.1.2 Назначение разработки** 4](#_Toc10301551)

[**1.1.3 Требования к программе** 4](#_Toc10301552)

[**1.1.4** **Требования к программной документации** 7](#_Toc10301553)

[**1.1.5 Порядок контроля и приёмки** 7](#_Toc10301554)

[**1.1.6 Календарный план работ** 8](#_Toc10301555)

[**1.2** **Обзор существующих решений задачи** 12](#_Toc10301556)

[**Глава II. Основная часть** 13](#_Toc10301557)

[**2.1** **Описание программы** 13](#_Toc10301558)

[**2.1.1 Общие сведения** 13](#_Toc10301559)

[**2.1.2 Функциональное назначение** 13](#_Toc10301560)

[**2.1.3 Структурная схема разработанного приложения** 14](#_Toc10301561)

[**2.1.4 Используемые технические средства** 16](#_Toc10301562)

[**2.1.5 Входные данные** 17](#_Toc10301563)

[**2.1.6 Выходные данные** 17](#_Toc10301564)

[**2.2** **Руководство программиста** 18](#_Toc10301565)

[**2.2.1 Общие сведения о программе** 18](#_Toc10301566)

[**2.2.2 Функциональная схема разработанного приложения** 19](#_Toc10301567)

[**2.2.3 Настройка программы** 22](#_Toc10301568)

[**2.2.4 Проверка программы** 22](#_Toc10301569)

[**2.2.5 Сообщения программисту** 25](#_Toc10301570)

[**2.3** **Руководство оператора** 28](#_Toc10301571)

[**2.3.1 Назначение программы** 29](#_Toc10301572)

[**2.3.2 Условие выполнение программы** 29](#_Toc10301573)

[**2.3.3 Выполнение программы** 30](#_Toc10301574)

[**2.3.4 Сообщения оператору** 45](#_Toc10301575)

[**ЗАКЛЮЧЕНИЕ** 48](#_Toc10301576)

[**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** 50](#_Toc10301577)

[**ПРИЛОЖЕНИЕ А** 51](#_Toc10301578)

[**ПРИЛОЖЕНИЕ Б** 189](#_Toc10301579)

# **ВВЕДЕНИЕ**

**Актуальность темы**. **В связи с** бурным ростом информационных технологий и увеличение объёма знаний, студентам зачастую не хватает времени для упорядочивания получаемых знаний и их дальнейшего использования, что напрямую влияет на время, необходимое для усвоения данной информации. **Таким образом** снижется уровень качества знаний, что на сегодняшний день для специалиста недопустимо.

Так же возникает необходимость повторения или изучения нового материала по определённым темам, но объём изучаемого материала не позволяет быстро и эффективно усваивать необходимую информацию, поэтому алгоритмизация способов изучения данного материала является первостепенной задачей.

**Известно, что** на сегодняшний день всё большую необходимость приобретает изучение английского языка, таким образом для подготовки специалистов в технической сфере английский язык так же необходим, как и другие, технические дисциплины, поэтому, **по мнению аналитиков**, применение новых, более эффективных способов изучения данного языка является ключевым методом достижения новых результатов.

Помимо глобальной тенденции повсеместного изучения английского языка существует еще одно важное обстоятельство, которое послужило причиной для создания электронного учебного пособия. В учебном процессе изучения английского языка в Московском техническом университете связи и информатики электронные учебные программы используются исключительно для проверки полученных знаний, а именно - программы для тестирования учащихся. **Отсюда следует, что** электронное пособие должно будет располагать новой возможностью - обучать пользователя опираясь на материал учебной программы.

**В связи с этим** мы можем сделать обоснованный вывод, что необходимость алгоритмизации способов изучения английского языка имеет первостепенную роль для обеспечения высокого качества знаний английского языка у студентов, подтверждая, что алгоритмизация способов изучения представляет собой **актуальную научную задачу**.

**Объектом исследования** являются способы эффективного изучения английского языка.

**Предметном исследования** является создание программы для изучения английского языка.

**Цель работы и задачи исследования**. Цель работы – алгоритмизация способов изучения нового материала по английскому языку, а также разработать программу, позволяющую эффективно обучатся и подготавливаться к экзаменам. Поставленная цель определила следующие основные задачи исследования:

1. Изучить учебные материалы и методические пособия по английскому языку, которые используются в Московском техническом университете связи и информатики.
2. Руководствуясь учебной программой, разработать электронное учебное пособие для подготовки к экзамену.

**Методы исследования**. Для решения поставленных задачи были использованы теоретические методы исследования Кожевниковой Т.В.

# **Глава I. Постановка задачи**

## **Техническое задание на разработку программного продукта**

### **1.1.1 Основания для разработки**

Основанием для данной работы служит техническое задание, выданное кафедрой “Иностранные языки”.

Исполнителями являются Сиротин Н. С. и Копиевский Н. Ю.

### **1.1.2 Назначение разработки**

Целью курсовой работы является создание электронного пособия, в котором будут использованы различные способы изучения нового материала по дисциплине “английский язык”, созданная в рамках курсовой работы программа предназначена для использования в “Московском техническом университете связи и информатики”.

### **1.1.3 Требования к программе**

**Требования к функциональным характеристикам**

1. Возможность доступа к материалам необходимым для подготовки к экзамену по иностранному языку (английский язык).
2. Возможность пройти подготовительное тестирование для указанного экзамена.
3. Наличие установочного файла.
4. Использование в структуре электронного пособия DLL-библиотек.
5. Наличие вступительной заставки приложения.
6. Наличие возможности идентификации пользователя.
7. Создать возможность записи результатов тестирования в специальную базу данных.
8. Предоставить возможность посмотреть результаты тестирования и составленный на основе результатов тестирования график.

**Требования к надёжности**

Для обеспечения надёжности необходимо следовать всем пунктам при установке программного обеспечения.

1. Программа должна работать с заданной информацией в соответствии с алгоритмом функционирования, в тех случаях когда программа будет получать данные в неверном формате или в другом случае невыполнения пользователем инструкций использования программы электронное пособие будет выводить сообщения об ошибках, поддерживать диалоговый режим в рамках предоставляемых пользователю возможностей.

**Условия эксплуатации**

Созданный программный продукт будет использоваться в Московском техническом университете связи и информатики, на кафедре иностранных языков.

Заданные характеристики функционирования должны обеспечиваться при условиях, которые определенны конкретным носителем, на котором хранятся данные электронного пособия.

Наиболее распространёнными физическими носителями, использующимися для распространения программного обеспечения, в настоящее время являются: жесткие диски, для которых оптимальным является функционирование при температуре воздуха от 5 до 40 °С и относительной влажности воздуха не более 80 %.

**Требование к составу и параметрам технических средств**

Минимальными системными требованиями к персональному компьютеру, на котором будет установлено программное обеспечение, являются:

Операционная система Windows 7 или более современная версия

Компьютер и процессор:

Процессор 1000 MHz или более, ОЗУ не менее 512 Мб

Жесткий диск:

Для установки необходимо 280 МБ

Разрешение экрана:

Потребуется монитор с разрешением 800х600 точек или более

Подключение к интернету:

Для установки и функционирования приложения не нужно подключение к сети интернет.

Рекомендуемыми системными требованиями к персональному компьютеру, на котором будет установлено программное обеспечение, являются:

Операционная система Windows 7 или более современная версия

Компьютер и процессор:

Процессор 1500 MHz или более, ОЗУ не менее 1 Гб

Жесткий диск:

Для установки необходимо 280 МБ

Разрешение экрана:

Потребуется монитор с разрешением 1024х768 точек или более

Подключение к интернету:

Для установки и функционирования приложения не нужно подключение к сети интернет.

**Требования программной совместимости**

Программное обеспечение должно легко инсталлироваться, функционировать и корректно работать при выполнении следующих требований:

* Наличие операционной системы Windows 7 или более современной версии.
* Для стабильной работы рекомендуется установить программу MS Access версии 12.0 (2007) или более поздняя версия.

**Требования к транспортировке и хранению**

Программа предоставлена на лазерном носителе. Программная документация - в электронном и печатном виде. На физическом носителе не должны находиться сторонние программы. Экземпляры электронного пособия на оптическом носителе информации будут в распоряжении кафедры иностранных языков.

**Специальные требования**

Программное обеспечение должно обеспечивать взаимодействие с пользователем по средствами графического интерфейса, разработанного согласно рекомендациям компании Microsoft.

Документация на принятое программное обеспечение должна содержать полную информацию, необходимую для пользователей и программистов, работающих с данной программой.

### **Требования к программной документации**

Основными документами, регламентирующими разрабатываемое программное обеспечение, являются документы, установленные единой системой программной документации (ЕСПД):

* Руководство оператора;
* Руководство программиста;
* Руководство системного администратора;
* Описание программы;
* Описание контрольного примера.

### **Порядок контроля и приёмки**

Приёмка результатов работы осуществляется в соответствии с планом, согласованным с руководителем курсовой работы. Данный план включает следующие пункты:

* Сдача технического задания и содержания пояснительной записки;
* Сдача программы;
* Защита курсовой работы.

Контроль и приёмка осуществляется на основе испытаний, при этом проверяется выполнение всех функций программы.

После предварительных испытаний программного продукта, заказчик имеет право тестировать программный модуль в течении определённого срока, после которого может принять работу по данному проекту или изложить причину отказа от принятия.

### **Календарный план работ**

В таблице 1 представлены этапы разработки программного продукта

Таблица 1 - Этапы разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Стадия разработки | Этапы работ | Содержание работ | Начало/Окончание |
| Проектирование | Постановка задачи | Задача была четко и конкретно сформулирована.  В соответствии с требованиями к программе были определены исходные данные и основа структуры электронного пособия | 04.03.19-11.03.19 |
|  | Создание Алгоритмов | Был намечен план решения задачи, определен путь её решения, весь результат проектирования был изложен на бумаге или в словесном виде | 04.03.19-11.03.19 |
| Реализация | Создание алгоритмов | Для решения задач было решено использовать измененные варианты уже существующих решений или совершенно новые алгоритмы, разработанные специально для электронного пособия | 11.03.19-25.03.19 |
|  | Запись программы на языке программирования | Все существующие на данном этапе алгоритмы, которые были записаны на алгоритмическом языке были перенесены в среду программирования Visual Studio на языке программирования C++ | 25.03.19-22.04.19 |
| Тестирование | Проверка | В целях экономии времени и систематизации процесса, в систему программы были временно внедрены специальные механизмы, которые позволили удостовериться в корректном выполнении программного кода электронного пособия | 22.04.19-29.04.19 |
|  | Отладка | Для исправления логических ошибок электронное пособие тестировалось путем многократного использования его полного функционала, а также с помощью метода тестирования отдельных модулей на ошибки, не связанные с синтаксисом программного кода | 29.04.19-05.05.19 |
| Внедрение | Эксплуатация | Программа стабильно функционирует, все поставленные задачи выполнены, программа готова для эксплуатации пользователем |  |

## **Обзор существующих решений задачи**

На данный момент в высшем учебном заведении “Московский технический университет связи и информатики” не существует аналогов к разрабатываемому электронному пособию, за исключение программ для тестирования учащихся. Недостатки данных тестирующих программ заключаются в том, что они не предоставляют возможность учащимся в данном учебном заведении, подготовится к предстоящим экзаменам, а лишь предполагает тестирование полученных знаний. Эксплуатируемые программы предлагают пользователю минимальный функционал и не совершенные методы тестирования, которые не обеспечивают объективную проверку знаний пользователя. Следовательно, возникает необходимость разработки данного программного обеспечения.

# **Глава II. Основная часть**

## **Описание программы**

### **Общие сведения**

Целью курсовой работы является алгоритмизация способов изучения нового материала по английскому языку, а также разработка программного обеспечения, позволяющее эффективно обучатся и подготавливаться к экзаменам. Для реализации цели, необходимо было решить следующие задачи:

1. Изучить учебные материалы и методические пособия по английскому языку, которые используются в Московском техническом университете связи и информатики.
2. Руководствуясь учебной программой, разработать электронное учебное пособие для подготовки к экзамену.
3. Разработать функцию, позволяющую изменять разрешение формы для разрешений экрана 800x600, 1024x766, 1366x766, 1920x1080.
4. Разработать функцию, позволяющую пользователю просматривать прогресс подготовки к экзамену.

В качестве среды разработки была выбрана среда программирования Microsoft Visual Studio 2017. В качестве языка программирования был выбран С++.

### **Функциональное назначение**

Разработанный программный продукт имеет следующие функциональные характеристики:

1. Возможность установки программы с использованием setup файла в формате “.exe”.
2. Возможность защиты от копирования.
3. Доступ к теоретическому материалу необходимому для прохождения экзамена по иностранному языку.
4. Возможность пройти предварительное/подготовительное тестирование.
5. Для пользователей, имеющих учетную запись, имеется доступ к персональной статистике, составленной на основе пройденного материала и результатах тестирования, для пользователей, не имеющих личную учетную запись предусмотрена возможность зарегистрировать учетную запись со своими данными.

### **Структурная схема разработанного приложения**

Структурная схема данного программного обеспечения представлена на рисунке 1. Данная схема показывает из каких блоков состоит программа. В каждом структурном блоке находится один или несколько функциональных блоков, исключением являются заставка, титульный лист и главное меню, данные блоки не несут функциональной нагрузки и предполагают своё включение в программу для расширения интерфейса пользователя.

Учётная запись (Вход) – предоставляет пользователю возможность войти в программу с использованием своего логина и пароля (при их отсутствии задействуется блок регистрации).

Регистрация – предоставляет пользователю возможность зарегистрировать собственную учётную запись в программе.

Настройки – предоставляют пользователю возможность выбрать размер форм программы.

Обучение – блок, включающий в себя возможность перехода от главного меню к отдельным блокам программы, а именно: к тестированию, к теоретической части (часть 1) – пересказы и теоретической части (часть 2) – резюме.

Тестирование – предоставляет пользователю возможность пройти тестирование для подготовки к экзамену по английскому языку (1 курс).

Теория и резюме – два отдельных блока, выполняющих схожие функции. Предоставляют пользователю ознакомиться с материалами, необходимыми для успешной подготовки к экзамену.

Личный кабинет – блок, включающий в себя возможность перехода от главного меню к отдельным блокам программы, а именно: к учётной записи, достижениям и календарю.

Учётная запись – предоставляет пользователю возможность увидеть свои данные, введённые во время регистрации.

Достижения – предоставляет пользователю возможность увидеть свой прогресс подготовки к экзамену по английскому языку (1 курс).

Календарь – предоставляет возможность пользователю увидеть сколько дней осталось до экзамена (дату прохождения экзамена пользователь, при желании, вводит сам).

Функциональные схемы программы представлены в главе 2.2.2.

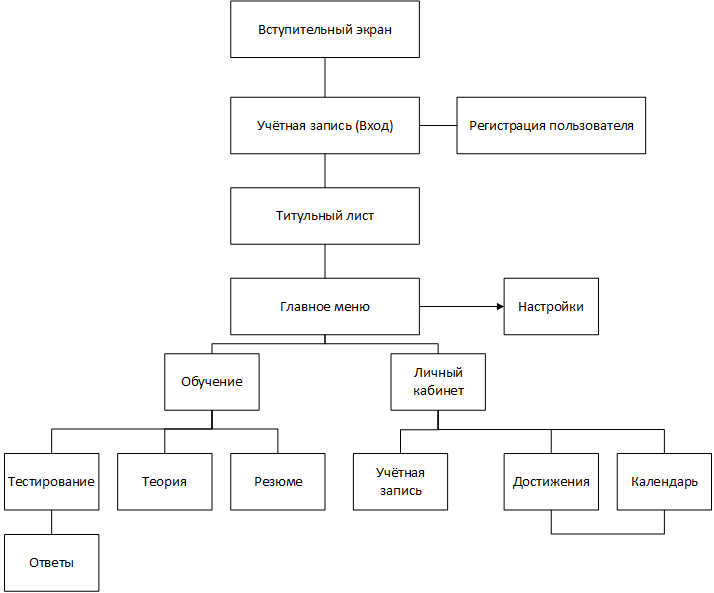


Рисунок 1 - Структурная схема разработанного приложения

### **Используемые технические средства**

Минимальными системными требованиями к персональному компьютеру, на котором будет установлено программное обеспечение, являются:

Операционная система Windows 7 или более современная версия

Компьютер и процессор:

Процессор 1000 MHz или более, ОЗУ не менее 512 Мб

Жесткий диск:

Для установки необходимо 280 МБ

Разрешение экрана:

Потребуется монитор с разрешением 800х600 точек или более

Подключение к интернету:

Для установки и функционирования приложения не нужно подключение к сети интернет.

Рекомендуемыми системными требованиями к персональному компьютеру, на котором будет установлено программное обеспечение, являются:

Операционная система Windows 7 или более современная версия

Компьютер и процессор:

Процессор 1500 MHz или более, ОЗУ не менее 1 Гб

Жесткий диск:

Для установки необходимо 280 МБ

Разрешение экрана:

Потребуется монитор с разрешением 1024х768 точек или более

Подключение к интернету:

Для установки и функционирования приложения не нужно подключение к сети интернет.

### **Входные данные**

Входными данными являются:

* Логин и пароль пользователя,
* Имя, фамилия, группа обучения пользователя,
* Варианты ответов для тестовых заданий.

### **Выходные данные**

Выходными данными являются:

* Ответы, введенные пользователем в процессе выполнения тестирования.
* Все данные, введённые пользователем для просмотра в личном кабинете.

## **Руководство программиста**

### **Общие сведения о программе**

Целью курсовой работы является алгоритмизация способов изучения нового материала по английскому языку, а также разработка программного обеспечения, позволяющее эффективно обучатся и подготавливаться к экзаменам. Для реализации цели, необходимо было решить следующие задачи:

1. Изучить учебные материалы и методические пособия по английскому языку, которые используются в Московском техническом университете связи и информатики.
2. Руководствуясь учебной программой, разработать электронное учебное пособие для подготовки к экзамену.
3. Разработать функцию, позволяющую изменять разрешение формы для разрешений экрана 800x600, 1024x766, 1366x766, 1920x1080.
4. Разработать функцию, позволяющую пользователю просматривать прогресс подготовки к экзамену.

Состав выполняемых функций разрабатываемого программного обеспечения, следующий:

* Проверка знаний пользователя,
* Обучение пользователя,
* Изменение размеров формы.

Рекомендуемыми системными требованиями к персональному компьютеру, на котором будет установлено программное обеспечение, являются:

Операционная система Windows 7 или более современная версия

Компьютер и процессор:

Процессор 1500 MHz или более, ОЗУ не менее 1 Гб

Жесткий диск:

Для установки необходимо 280 МБ

Разрешение экрана:

Потребуется монитор с разрешением 1024х768 точек или более

Подключение к интернету:

Для установки и функционирования приложения не нужно подключение к сети интернет.

Программное обеспечение должно легко инсталлироваться, функционировать и корректно работать при выполнении следующих требований:

* Наличие операционной системы Windows.

В качестве среды разработки была выбрана среда программирования Microsoft Visual Studio 2017. В качестве языка программирования был выбран С++.

### **Функциональная схема разработанного приложения**

**Функциональная схема**

На функциональной схеме стрелками указаны функции входящие в структурный блок. Каждый функциональный блок привязан к одному или нескольким структурным. Функциональная схема программы представлена на рисунке 2. Жирным шрифтом выделены структурные блоки программы, объединяющие функциональные.

Ниже представлено описание функций каждого блока. Программный код всех функций находится в приложении А.

Вход – функциональный блок, в котором происходит проверка наличия пользователя в базе данных и последующий его вход в систему.

Регистрация – функциональный блок, проверяющий на повторное существование логина пользователя в базе данных и, в случае не совпадения, запись пользователя в базу данных.

Настройки – функциональный блок, задающий размер каждой формы.

Теория 1 – функциональный блок вывода данных по пересказам пользователю.

Теория 2 – функциональный блок вывода данных по написанию резюме пользователю.

Тест 1 – функциональный блок прохождения тестирования первого типа (выбор перевода фразы).

Тест 2 – функциональный блок прохождения тестирования второго типа (ввод правильного ответа вручную).

Тест 3 – функциональный блок прохождения тестирования третьего типа (выбор правильного ответа среди предложенных).

Ответы – функциональный блок вывода данных тестирования и их запись в базу данных.

Учётная запись – функциональный блок для вывода данных пользователя.

Достижения – функциональный блок проверки пройдённых тестов, чтения теории и других достижения пользователя и их вывод на форму для последующего ознакомления с данной информацией.

Календарь – функциональный блок показа информации об остатке дней до экзамена по английскому языку (1 курс).

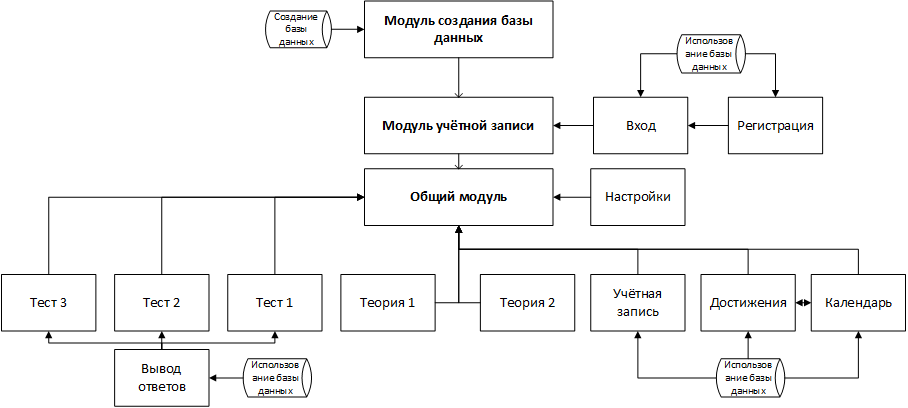


Рисунок 2 - Функциональная схема приложения

**Архитектура классов**

В разработанном приложении существует несколько функциональных классов: AllTest\_Module, AllTest, Login\_Module и ChangeForm\_Module. Первые два класса используются в тестирующей части программы и предполагают использование внутренних функций только для формы AllTests.

Класс AllTest\_Module обладает двумя внутренними классами PhraseTranslate\_Module и RadioButton\_Module, которые задают функции для определённых вопросов тестирования.

Класс AllTest задаёт функции, выполняемые во всей тестирующей части, а именно переключения trackbar, изменения видимости форм, события загрузки формы, вывод ответов.

Класс Login\_Module предполагает использование на форме Login, для регистрации пользователя и его дальнейшего входа в программу.

Класс ChangeForm\_Module используется на всех формах для изменения размеров формы.

### **Настройка программы**

В настройках программы предоставляется возможность выбрать необходимое разрешения экрана для более комфортной работы. Выбор разрешения экрана производится из 4 возможных состояний 800x600, 1024x768, 1366x768, 1920x1080 пикселей.

ВНИМАНИЕ: при установке на системный диск программу следует запускать от имени администратора, в противном случае работоспособность программы будет нарушена.

### **Проверка программы**

**Объект испытаний**

Объектом испытаний является разработанное программное обеспечение

**Описание тестирование программы**

Для тестирования программного обеспечения на ранних этапах был использован метод ручного контроля. При данном типе тестирования создавалась группа специалистов: авторы, специалист по тестирования, координатор. Участникам был выдан листинг программы и спецификация на разработку. Специалисты ответили на ряд вопросов, связанных с тестированием программы:

1. Все ли переменные инициализированы?
2. Не превышен ли размер массивов/строк?
3. Правильно ли используется работа с файлами?
4. Правильный ли приоритет выполнения алгоритмов?
5. Корректно ли выполняется вычисление переменных разных типов?
6. Существуют ли циклы, которые не будут выполнятся из-за нарушения условий выполнения?

В ходе данного тестирования были обнаружены следующие ошибки:

* В частях тестирования запись массива была превышена.
* Тестирование предполагало неправильные приоритеты выполнения кода.
* При работе с файлом, после выполнения процедуры, не происходило завершение работы с файлом.

Других ошибок на данном этапе не было обнаружено.

**Тестирование модулей**

Для тестирования модулей программы был применен метод комбинированного тестирования. Для модулей верхнего уровня тестирование нисходящее, для модулей низшего уровня – восходящее. Данный тип тестирования позволил обнаружить определённое количество ошибок.

**Восходящее тестирование**

Проблема №1. Найдена в общем модуле DLL библиотек. Ошибка кодирования. Фатальная.

При сборке модуля происходят ошибки, которые препятствуют передаче возвращаемых значений в другие модули, что препятствует запуску программы.

Состояние: Закрыта. Исправлена.

Был изменён способ передачи значений между модулями.

Проблема №2. Найдена в модуле тестирования. Предложение.

Предложение объединить два модуля (PhraseTranslate и RadioButton) для более рационального использования ресурсов программы.

Состояние: Закрыто. Предложение принято.

Модули PhraseTranslate и RadioButton были объединены.

Проблема №3. Найдена в модуле тестирования. Ошибка проектирования. Незначительная.

При объединении модулей возникли логические ошибки проектирования при распределении классов.

Состояние: Закрыто. Исправлена.

Классы были перераспределены.

Проблема №4. Найдена в теоретическом модуле. Ошибка кодирования. Серьёзная.

После сборки модуля теряется часть значений строк в связи с ошибками кодировки знаков, но возможность работы с содержимым модуля сохраняется.

Состояние: Закрыта. Исправлена.

Были изменены знаки, вызывавшие нарушение чтение файла.

Проблема №5. Найдена в модуле тестирования. Ошибка проектирования. Серьёзная.

При проектировании проекта не учитывалась программная возможность передачи данных в функциональный модуль ответов.

Состояние: Закрыта. Исправлена.

В модуль была добавлена функция для передачи данных в Answers.

**Нисходящее тестирование**

Проблема №1. Найдена в основной части проекта. Ошибка проектирования. Серьёзная.

При построении проекта теряется часть данных из модулей AllTest и Login.

Состояние: Закрыта. Исправлена.

Использованы глобальные переменные, предотвращающие потерю данных между модулями.

В результате проверки программы был исправлен ряд проблем. На данный момент программное обеспечение исправно функционирует.

**Дополнительные возможности**

Дополнительными функциями программы являются:

* Возможность использования учётной записи и просмотра достижений
* Просмотр прогресса подготовки к экзамену
* Создания выборочного тестирования
* Просмотр плана подготовки в календаре

### **Сообщения программисту**

Ниже представлен список и изображения сообщений, выводимых программой.

**Вход и регистрация**

Данное сообщение появляется при нажатии кнопки авторизация с пустыми полями логина и пароля (рис. 3).

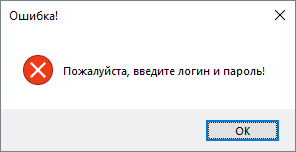


Рисунок 3 - Ошибка логина и пароля

Сообщение об ошибке, появляющееся при неправильно введённом пароле (рис. 4).

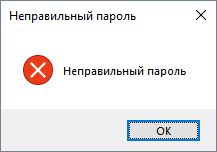


Рисунок 4 - Неправильный пароль

Сообщение, появляющееся при правильно введённом логине и пароле (рис. 5).

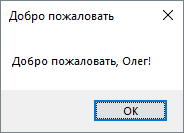


Рисунок 5 - Правильный логин и пароль

Сообщение, появляющееся при отсутствии данных хотя бы в одном поле регистрации (рис. 6).

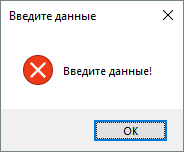


Рисунок 6 - Ошибка ввода данных

Сообщение об ошибке, высвечивающееся при регистрации пользователя с уже существующим логином (рис. 7).

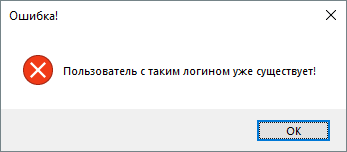


Рисунок 7 - Ошибка регистрации пользователя

**Учётная запись**

Сообщение ошибки ввода данных пользователем (число и месяц) (рис. 8).

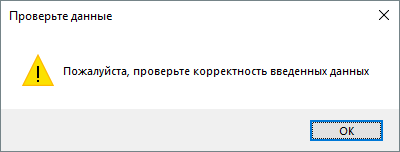


Рисунок 8 - Ошибка данных

**Тестирование**

Сообщение-вопрос, появляющееся при нажатии кнопки “Завершить тест досрочно”. При нажатии варианта “Да”, тестирование прекращается (рис. 9).

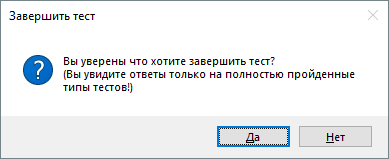


Рисунок 9 - Сообщение досрочного завершения теста

Сообщение-предупреждение, появляющееся при добавлении лишнего элемента в тестировании на составление фразы (рис. 10).

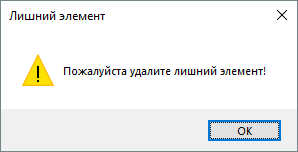


Рисунок 10 - Предупреждение о лишнем элементе

Сообщение, появляющееся при нажатии пользователем на кнопку “Ответить” при невыбранном варианте ответа (рис. 11). Появляется в каждом типе тестирования.

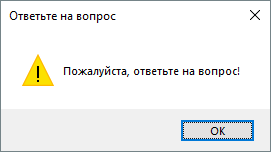


Рисунок 11 - Сообщение ответа на вопрос

Сообщение, показывающее количество правильных ответов (рис. 12).

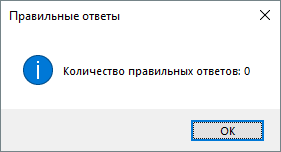


Рисунок 12 - Количество правильных ответов

## **Руководство оператора**

### **2.3.1 Назначение программы**

Целью курсовой работы является алгоритмизация способов изучения нового материала по английскому языку, а также разработка программного обеспечения, позволяющее эффективно обучатся и подготавливаться к экзаменам. Для реализации цели, необходимо было решить следующие задачи:

1. Изучить учебные материалы и методические пособия по английскому языку, которые используются в Московском техническом университете связи и информатики.
2. Руководствуясь учебной программой, разработать электронное учебное пособие для подготовки к экзамену.

Разработанный программный продукт имеет следующие функциональные характеристики:

1. Возможность установки программы с использованием setup файла.
2. Возможность защиты от копирования.
3. Возможность прочитать теоретический материал необходимый для прохождения экзамена по иностранному языку.
4. Возможность пройти предварительное/подготовительное тестирование.

### **Условие выполнение программы**

Минимальными системными требованиями к персональному компьютеру, на котором будет установлено программное обеспечение, являются:

Операционная система Windows 7 или более современная версия

Компьютер и процессор:

Процессор 1000 MHz или более, ОЗУ не менее 512 Мб

Жесткий диск:

Для установки необходимо 280 МБ

Разрешение экрана:

Потребуется монитор с разрешением 800х600 точек или более

Подключение к интернету:

Для установки и функционирования приложения не нужно подключение к сети интернет.

Рекомендуемыми системными требованиями к персональному компьютеру, на котором будет установлено программное обеспечение, являются:

Операционная система Windows 7 или более современная версия

Компьютер и процессор:

Процессор 1500 MHz или более, ОЗУ не менее 1 Гб

Жесткий диск:

Для установки необходимо 280 МБ

Разрешение экрана:

Потребуется монитор с разрешением 1024х768 точек или более

Подключение к интернету:

Для установки и функционирования приложения не нужно подключение к сети интернет.

Программное обеспечение должно легко инсталлироваться, функционировать и корректно работать при выполнении следующих требований:

* Наличие операционной системы Windows.

### **Выполнение программы**

Для того чтобы начать работу с электронным пособием запустите приложение SirotinKopievskii.exe, расположенном в корневой папке программы “C:\Program Files (x86)\MTUCI\SirotinKopievskii”, или нажмите на создавшийся после выполнения установки приложения с помощью setup-файла ярлык программы электронного пособия на рабочем столе.

ВНИМАНИЕ: при установке на системный диск программу следует запускать от имени администратора, в противном случае работоспособность программы будет нарушена.

С запуском программы появляется загрузочная форма, на которой находится название электронного пособия и отображается статус готовности программы, запуск программы изображен на рисунке 13.

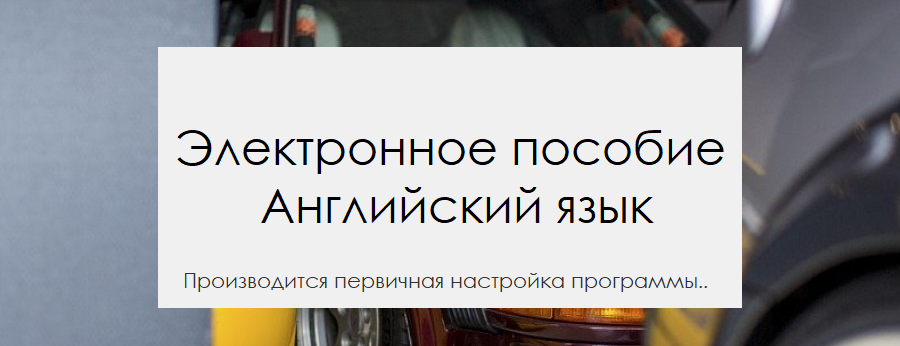


Рисунок 13 - Загрузочная форма электронного пособия

При первом запуске программы, произойдёт её автоматическая настройка. Последующие запуски программы сопровождаются инициализацией формы авторизации пользователя, она изображена на рисунке 14.

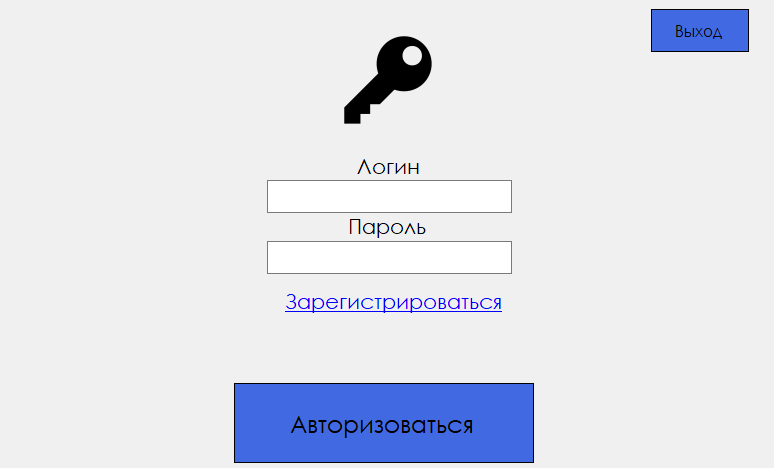


Рисунок 14 - Интерфейс формы авторизации пользователя

При вводе данных в поле “Пароль” символы введенных данных будут визуально изменены на символы “\*”, эффект отображен на рисунке 15.

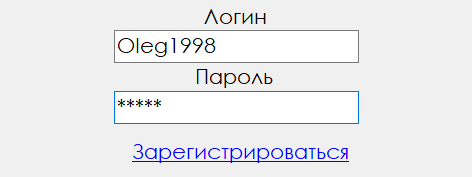


Рисунок 15 - Поле с данными пароля визуально скрыты символами “\*”

В случае если у пользователя нет учетной записи он может создать её, нажав на “Зарегистрироваться”, для регистрации на этой же форме будет отображен интерфейс, в котором пользователь введет все необходимые персональные данные (рис. 16).

Программа обладает защитой от дублирования учетных записей, поэтому в случае, если пользователь введет в поле регистрации логин пользователя, который уже зарегистрирован в системе, то программа оповестит его с помощью сообщения MessageBox, в таком случае пользователю будет необходимо зарегистрировать другой логин. Пример MessageBox показан на рисунке 18.

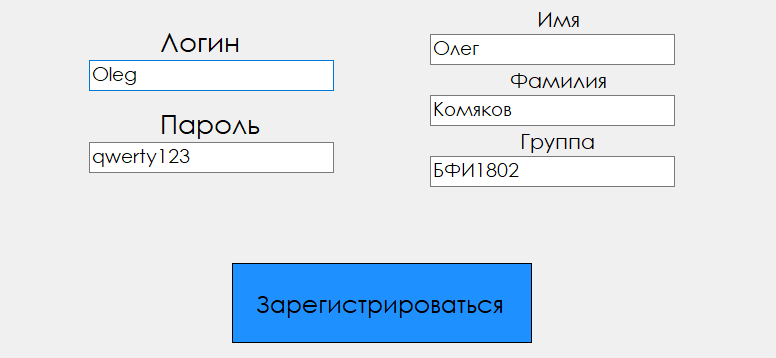


Рисунок 16 - Пользователь заполнил анкету для регистрации

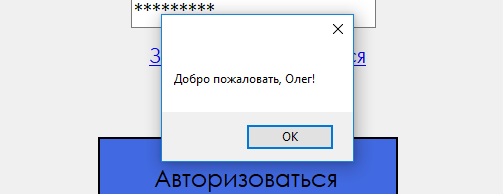


Рисунок 17 - MessageBox с приветствием пользователя

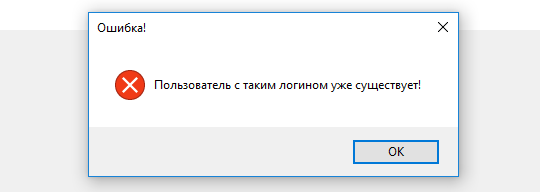


Рисунок 18 - Предупреждение пользователя о том, что логин уже используется

Во время первого запуска для пользователя выводится специальное окно, в котором он может выполнить первоначальную настройку изображения (рис. 19), разрешение приложения можно будет поменять в дальнейшем по желанию пользователя любое количество раз. При нажатии кнопки “ОК” без выбора разрешения, выберется стандартное разрешение 800x600.

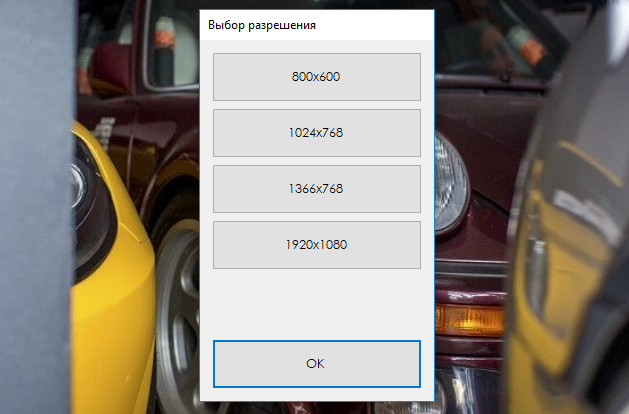


Рисунок 19 - Меню выбора разрешения приложения

После нажатия клавиши “ОК” пользователь будет перемещен на титульный лист программы (рис. 20), после нажатия на клавишу “Начать работу” для пользователь получит доступ ко всем функциям программы.

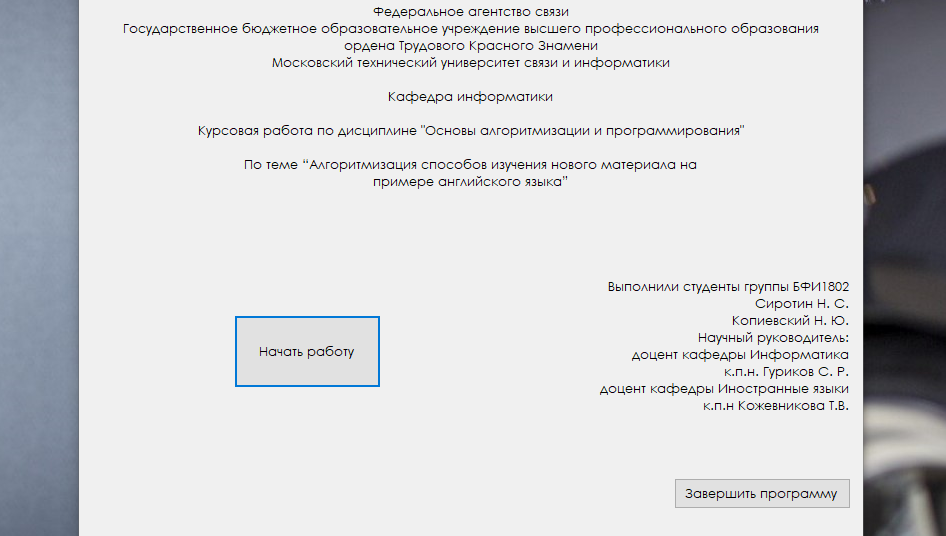


Рисунок 20 - Титульная форма электронного пособия

Находясь в главном меню электронного пособия, пользователь может получить доступ к теоретической части, тестам, настройкам программы (рис. 21).



Рисунок 21 - Внешний вид главного меню

**Теоретическая часть**

После нажатия на элемент “Обучение” для пользователя откроется интерфейс навигации по учебным материалам пособия по английскому языку. Перемещение осуществляется нажатием на интересующий учащегося элемент “Тестирование”, “Теория” или “Резюме”, в случае если оператору необходимо вернуть исходный вид главного меню, то ему нужно повторно нажать на элемент “Обучение” (рис. 22).

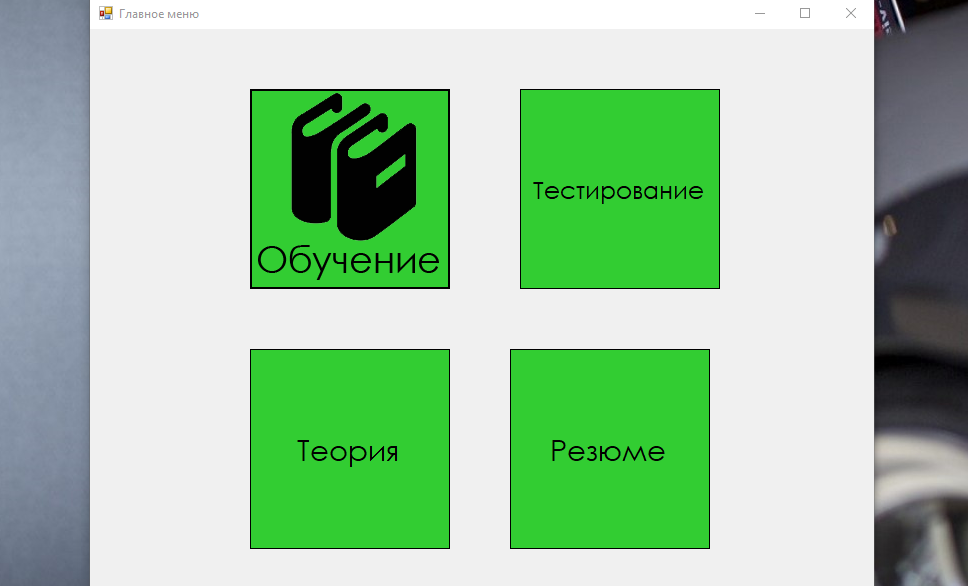


Рисунок 22 - Вкладка “Обучение” в главном меню

Пользователь может посетить раздел “Теория”, где находятся материалы для подготовки текстов для пересказа в рамках экзамена по английскому языку (1 курс). Пользователь может переключаться между шестью текстами с помощью элемента управления ComboBox (рис. 23).

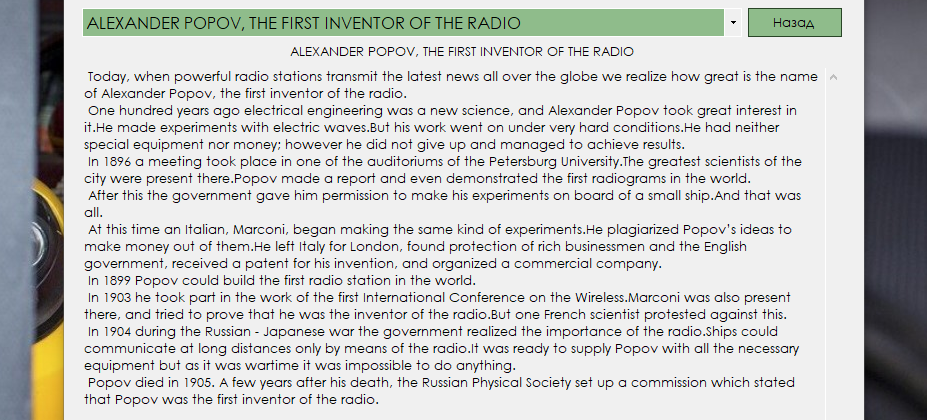


Рисунок 23 - Внешний вид вкладки “Теория”

Для подготовки к написанию резюме доступен раздел “Резюме”, пользователь может просмотреть готовый варианты-примеры резюме, изображен на рисунке 24, или ознакомиться и повторить структуру резюме, рисунок 25.

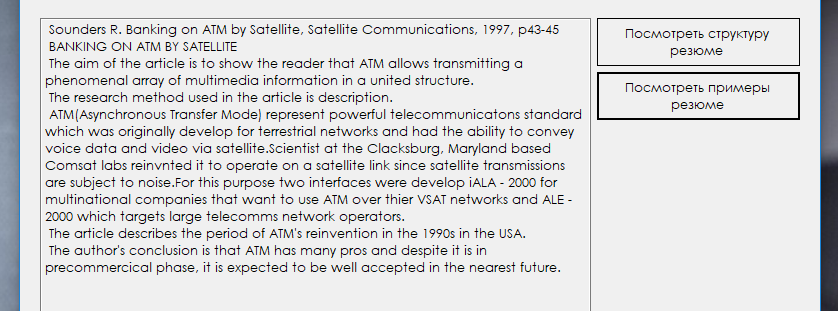


Рисунок 24 - Готовый пример резюме

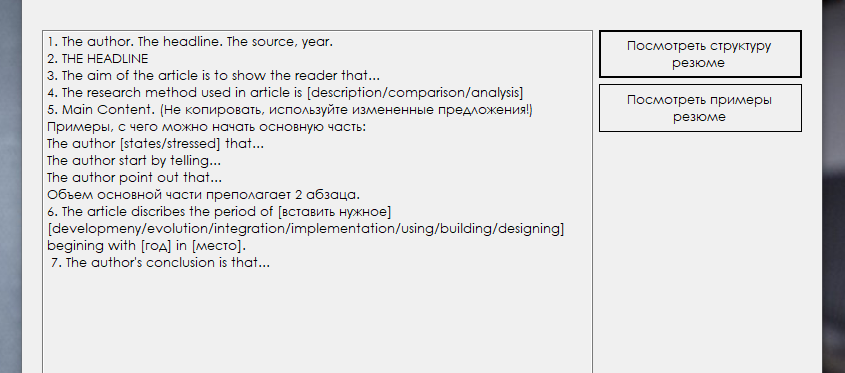


Рисунок 25 - Структура написания резюме

**Тестирование**

После перехода к “Тестирование” у пользователя будет возможность выбора варианта тестирования. Интерфейс меню составления теста представлен на рисунке 26. Пользователь не может начать тестирование, не выбирая ни одного из возможных вариантов тестирования. Кнопка начать тест активируется только если выбран минимум один пункт (рис. 27).

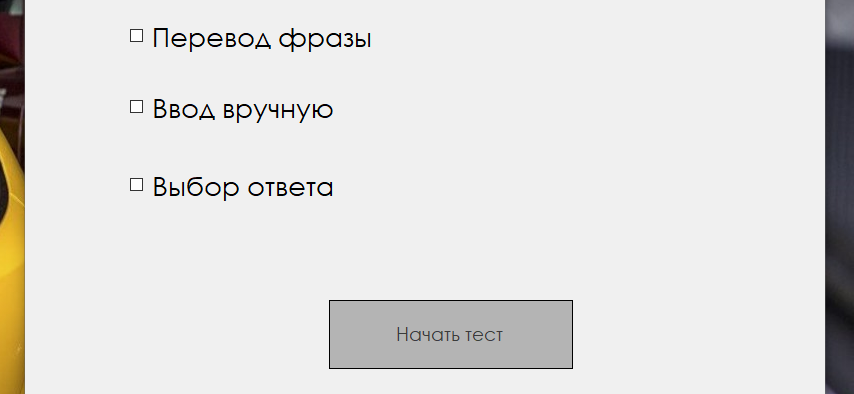


Рисунок 26 - Пользователь еще не выбрал состав тестирования

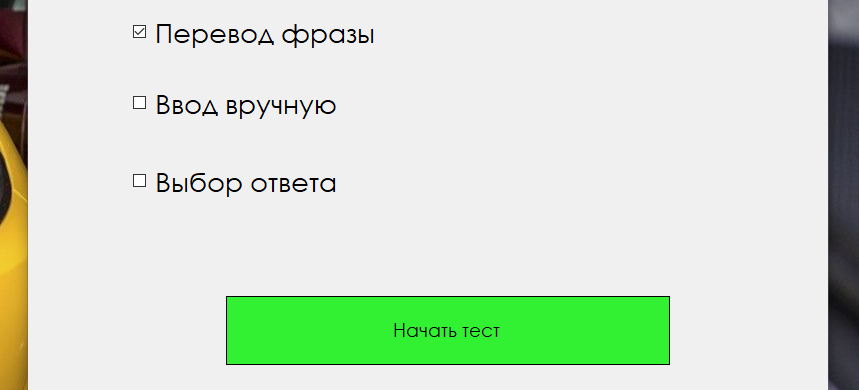


Рисунок 27 - Пользователь выбрал пункт “Перевод фразы”

Весь интерфейс тестовых заданий располагается на элементах Panel. Программа отображает нужный пользователю интерфейс в зависимости от состава, выбранного им ранее тестирования (рис. 28).

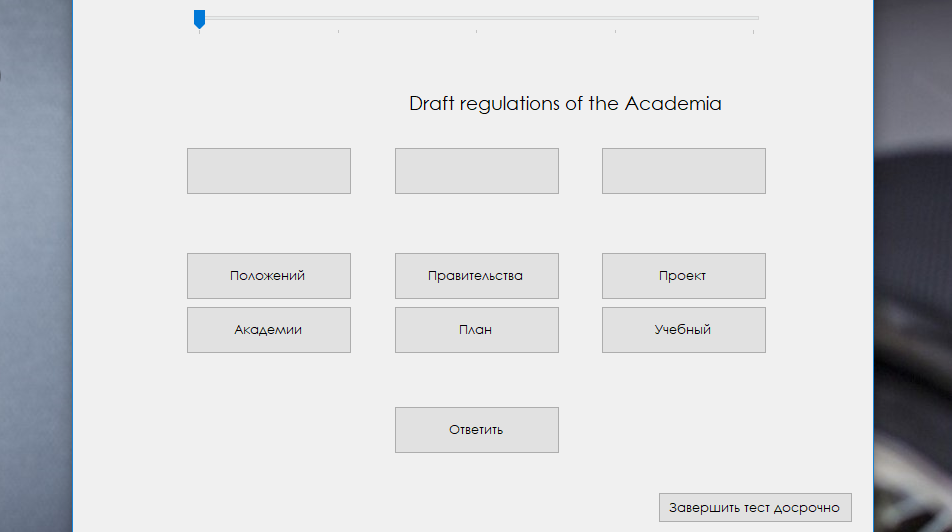


Рисунок 28 - Внешний вид тестирования “Перевод фразы”

Для выполнения задания составьте перевод определенной фразы, написанной в верхней части экрана на английском языке заполняя три пустые поля фразой, которую нужно составить из предложенных ниже вариантов, при нажатии на один из них он добавляется в первое свободное поле слева, очистить поле можно нажатием на это самое поле.

Для перемещения по заданиям тестирования перетаскивайте ползунок элемента TrackBar в верхней части. Каждому положению соответствует определенное задание (рис. 29).

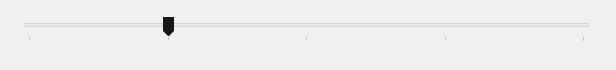


Рисунок 29 – TrackBar осуществляет переход к другому заданию

После того как пользователь сделает выполнит пять заданий одного типа он может перейти к следующему типу заданий нажав на “Следующий вопрос” в нижней части экрана (рис. 30). Для пользователя будет показана сводка о количестве правильных ответов в окне MessageBox (рис. 31).

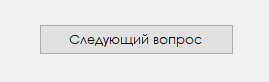


Рисунок 30 - Кнопка “Следующий вопрос”, появившаяся после выполнения пяти заданий

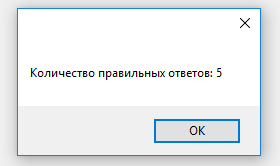


Рисунок 31 - Оповещение для пользователя

В тестовой программе предусмотрена возможность завершить тестирование досрочно, эта функция доступна с первого задания тестирования (рис. 32). Перед тем как завершить тест программа еще раз убедиться в намерениях пользователя в диалоговой форме (рис. 33).

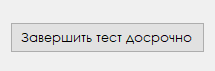


Рисунок 32 - Кнопка “Завершить тест досрочно”

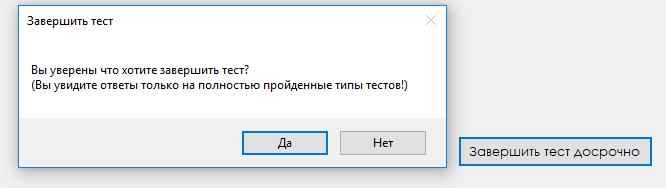


Рисунок 33 – MessageBox для подтверждения намерений пользователя

Среди тестовых заданий могут быть задания типа “Ввод вручную”, пользователю требуется написать одно слово в пустое поле - краткую формулировку фразу представленной выше на английском языке, ответ аналогично - на английском языке. Для этого задания существуют множественные верные варианты ответа, как и возможность вводить ответ с заглавной или строчной буквы (рис. 34).

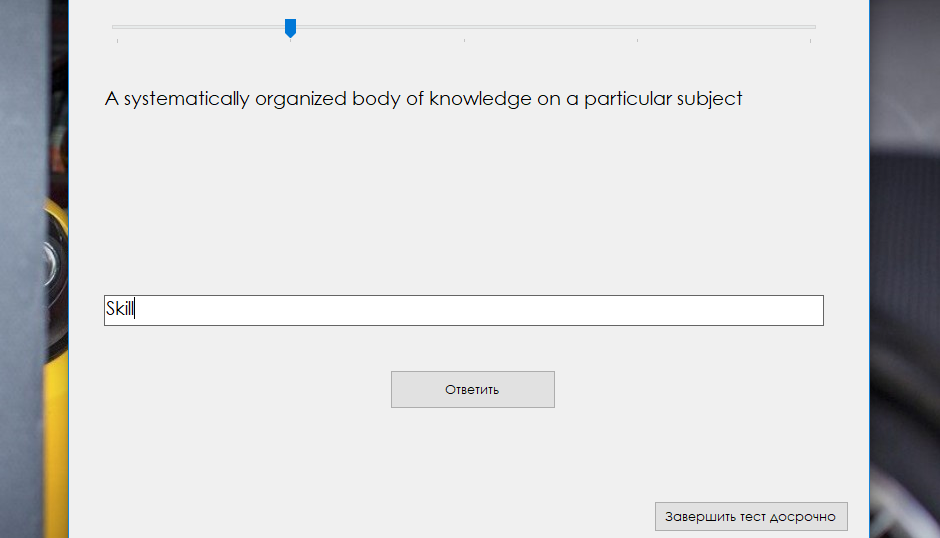


Рисунок 34 - Внешний вид задания “Ввод вручную”

После завершения тестирования пользователь может ознакомиться со своими с результатами тестирования, они будут отображены в элементе DataGridView, вывод ответов изображен на рисунке 35.

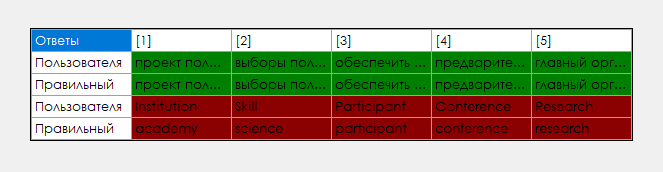


Рисунок 35 - Внешний вид элемента DataGridView

**Кабинет**

Для расширения функционала электронного пособия была добавлена возможность отслеживать персональный прогресс, все данные о котором собраны во вкладке “Кабинет”, интерфейс позволяет получить доступ к учетной записи, где содержится персональная информация о пользователе, календарь, в котором отслеживается дата проведения экзамена и достижения, система достижений вознаграждает пользователя за продвижение в учебе (рис. 36).

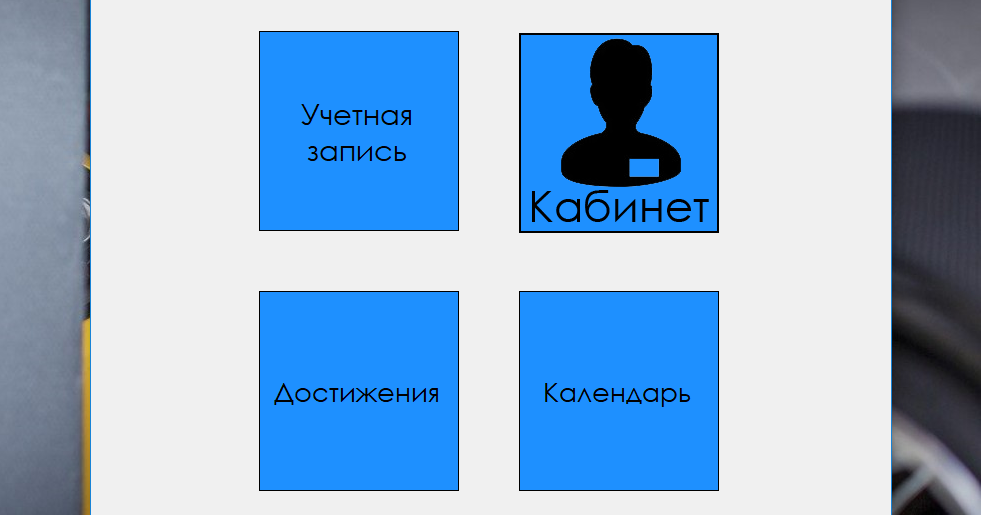


Рисунок 36 - Внешний вид вкладки “Кабинет”

После перемещения в “Достижения”, интерфейс изображен на рисунке 37, пользователю будет доступен статус его достижений и возможность ввести дату сдачи экзамена для того чтобы воспользоваться вкладкой “Календарь”.

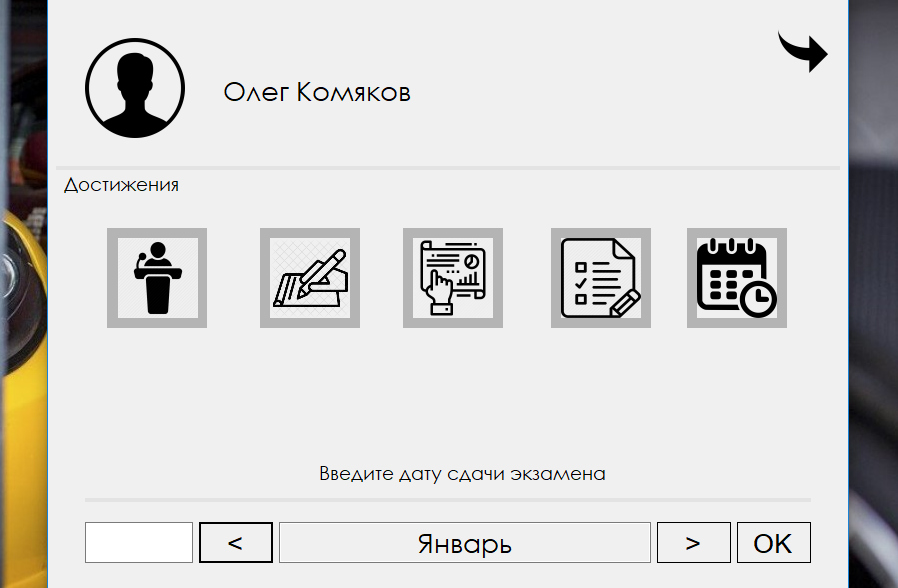


Рисунок 37 - Внешний вид вкладки “Достижения”

По мере изучения программы и выполнения заданий пользователю будут начисляться “очки” достижений, каждому достижению соответствует своя часть программы (рис. 38).

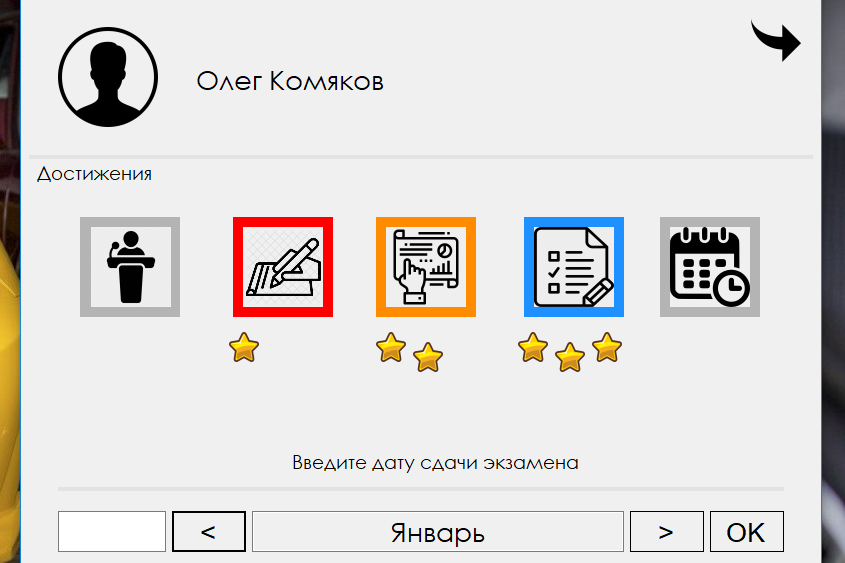


Рисунок 38 - Отображение “очков” достижений

Пользователь может ввести дату сдачи экзамена, в пустое поле требуется ввести число сдачи экзамена и выбрать с помощью кнопок месяц сдачи экзамена, подтвердить с помощью кнопки “ОК” (рис. 39, 40).

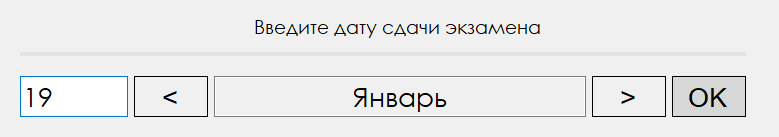


Рисунок 39 - Пользователь ввел дату сдачи экзамена

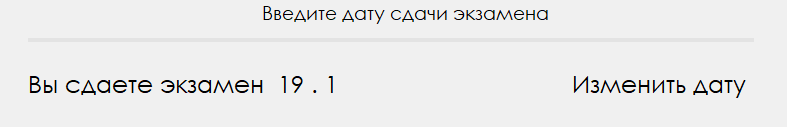


Рисунок 40 - Пользователь подтвердил сдачу экзамена

После того как пользователь ввел дату сдачи экзамена во вкладке “Календарь” отмечается соответствующий день (рис. 41).

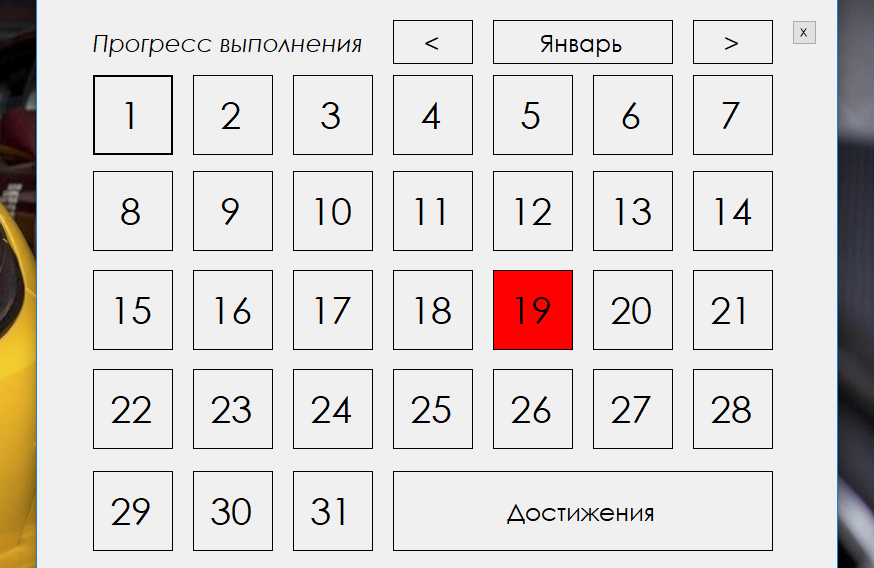


Рисунок 41 - Внешний вид вкладки “Календарь”

### **Сообщения оператору**

Ниже представлен список и изображения сообщений, выводимых программой.

**Вход и регистрация**

Данное сообщение появляется при нажатии кнопки авторизация с пустыми полями логина и пароля (рис. 42).

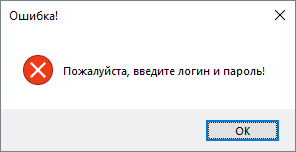


Рисунок 42 - Ошибка логина и пароля

Сообщение об ошибке, появляющееся при неправильно введённом пароле (рис. 43).

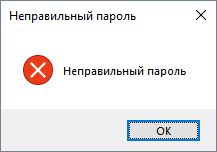


Рисунок 43 - Неправильный пароль

Сообщение, появляющееся при правильно введённом логине и пароле (рис. 44).

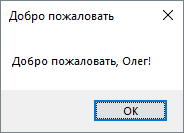


Рисунок 44 - Правильный логин и пароль

Сообщение, появляющееся при отсутствии данных хотя бы в одном поле регистрации (рис. 45).

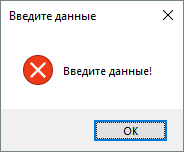


Рисунок 45 - Ошибка ввода данных

Сообщение об ошибке, высвечивающееся при регистрации пользователя с уже существующим логином (рис. 46).

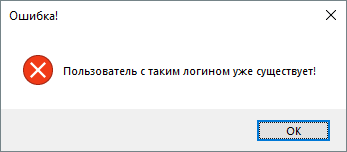


Рисунок 46 - Ошибка регистрации пользователя

**Учётная запись**

Сообщение ошибки ввода данных пользователем (число и месяц) (рис. 47).

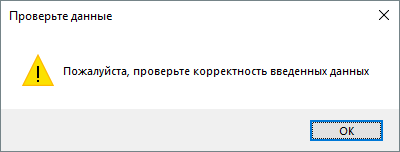


Рисунок 47 - Ошибка данных

**Тестирование**

Сообщение-вопрос, появляющееся при нажатии кнопки “Завершить тест досрочно”. При нажатии варианта “Да”, тестирование прекращается (рис. 48).

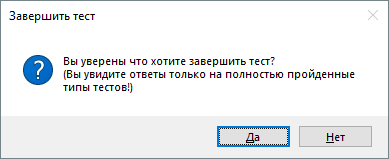


Рисунок 48 - Сообщение досрочного завершения теста

Сообщение-предупреждение, появляющееся при добавлении лишнего элемента в тестировании на составление фразы (рис. 49).

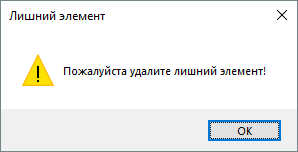


Рисунок 49 - Предупреждение о лишнем элементе

Сообщение, появляющееся при нажатии пользователем на кнопку “Ответить” при невыбранном варианте ответа (рис. 50). Появляется в каждом типе тестирования.

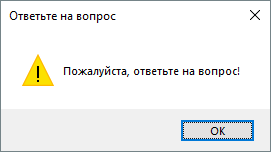


Рисунок 50 - Сообщение ответа на вопрос

Сообщение, показывающее количество правильных ответов (рис. 51).

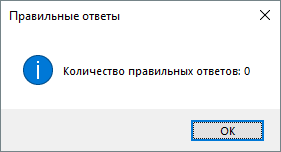


Рисунок 51 - Количество правильных ответов

# **ЗАКЛЮЧЕНИЕ**

В рамках курсового проекта был разработан программный продукт “СиротинКопиевский” предназначенный для подготовки к экзамену по английскому языку (1 курс) в Московском техническом университете связи и информатики для использования на кафедре иностранных языков. В результате выполнения работы можно сделать следующие выводы:

1. При разработке программного обеспечения был пройден полный цикл проектирования программы: от постановки задачи до её успешного тестирования и сборки финального решения.
2. Разработанный программный продукт позволяет:

* Установить программу с использованием setup файла.
* Получить доступ к теоретическому материалу необходимому для прохождения экзамена по иностранному языку.
* Возможность пройти предварительное/подготовительное тестирование.
* Возможность выбора размеров формы.
* Возможность просмотреть прогресс подготовки к экзамену.

Таким образом, подводя итоги курсовой работы следует отметить, что были достигнуты все поставленные цель и задачи.

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Кожевникова Т.В. Английский язык для университетов и институтов связи/ Кожевникова Т.В. – Москва: КНОРУС, 2012. - 368 с.

# **ПРИЛОЖЕНИЕ А**

**AchivmentsForm.h**

#pragma once

#include "VAR.h"

#include "TestsHeaders.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

using namespace Data::OleDb;

using namespace Obj\_Space;

/// <summary>

/// Сводка для AchivmentsForm

/// </summary>

public ref class AchivmentsForm : public System::Windows::Forms::Form

{

public:

AchivmentsForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~AchivmentsForm()

{

if (components)

{

delete components;

}

}

protected:

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::PictureBox^ pictureBox1;

private: System::Windows::Forms::Panel^ panel1;

private: System::Windows::Forms::Panel^ panel2;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::PictureBox^ pictureBox2;

private: System::Windows::Forms::PictureBox^ pictureBox3;

private: System::Windows::Forms::PictureBox^ pictureBox4;

private: System::Windows::Forms::PictureBox^ pictureBox5;

private: System::Windows::Forms::PictureBox^ pictureBox6;

private: System::Windows::Forms::PictureBox^ pictureBox7;

private: System::Windows::Forms::PictureBox^ pictureBox8;

private: System::Windows::Forms::PictureBox^ pictureBox9;

private: System::Windows::Forms::PictureBox^ pictureBox10;

private: System::Windows::Forms::PictureBox^ pictureBox12;

private: System::Windows::Forms::PictureBox^ pictureBox13;

private: System::Windows::Forms::PictureBox^ pictureBox14;

private: System::Windows::Forms::PictureBox^ pictureBox15;

private: System::Windows::Forms::PictureBox^ pictureBox16;

private: System::Windows::Forms::PictureBox^ pictureBox17;

private: System::Windows::Forms::PictureBox^ pictureBox18;

private: System::Windows::Forms::PictureBox^ pictureBox19;

private: System::Windows::Forms::PictureBox^ pictureBox20;

private: System::Windows::Forms::PictureBox^ pictureBox21;

private: System::Windows::Forms::PictureBox^ pictureBox22;

private: System::Windows::Forms::PictureBox^ pictureBox23;

private: System::Windows::Forms::PictureBox^ pictureBox24;

private: System::Windows::Forms::PictureBox^ pictureBox25;

private: System::Windows::Forms::Panel^ panel3;

private: System::Windows::Forms::Panel^ panel4;

private: System::Windows::Forms::Panel^ panel5;

private: System::Windows::Forms::Panel^ panel6;

private: System::Windows::Forms::Panel^ panel7;

private: System::Windows::Forms::ToolTip^ toolTip1;

private: System::Windows::Forms::Timer^ timer1;

private: System::Windows::Forms::Panel^ panel8;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::Panel^ panel9;

private: System::Windows::Forms::Timer^ ComfortPlus;

private: System::Windows::Forms::Timer^ timer2;

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->panel2 = (gcnew System::Windows::Forms::Panel());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox3 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox4 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox5 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox6 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox7 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox8 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox9 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox10 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox12 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox13 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox14 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox15 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox16 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox17 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox18 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox19 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox20 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox21 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox22 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox23 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox24 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox25 = (gcnew System::Windows::Forms::PictureBox());

this->panel3 = (gcnew System::Windows::Forms::Panel());

this->panel4 = (gcnew System::Windows::Forms::Panel());

this->panel5 = (gcnew System::Windows::Forms::Panel());

this->panel6 = (gcnew System::Windows::Forms::Panel());

this->panel7 = (gcnew System::Windows::Forms::Panel());

this->toolTip1 = (gcnew System::Windows::Forms::ToolTip(this->components));

this->timer1 = (gcnew System::Windows::Forms::Timer(this->components));

this->panel8 = (gcnew System::Windows::Forms::Panel());

this->label5 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->panel9 = (gcnew System::Windows::Forms::Panel());

this->ComfortPlus = (gcnew System::Windows::Forms::Timer(this->components));

this->timer2 = (gcnew System::Windows::Forms::Timer(this->components));

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox2))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox3))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox4))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox5))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox6))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox7))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox8))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox9))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox10))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox12))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox13))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox14))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox15))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox16))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox17))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox18))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox19))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox20))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox21))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox22))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox23))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox24))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox25))->BeginInit();

this->panel8->SuspendLayout();

this->panel9->SuspendLayout();

this->SuspendLayout();

//

// textBox2

//

this->textBox2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox2->Location = System::Drawing::Point(281, 10);

this->textBox2->Margin = System::Windows::Forms::Padding(4);

this->textBox2->Name = L"textBox2";

this->textBox2->ReadOnly = true;

this->textBox2->Size = System::Drawing::Size(495, 49);

this->textBox2->TabIndex = 0;

this->textBox2->TabStop = false;

this->textBox2->Text = L"Январь";

this->textBox2->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;

//

// button1

//

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button1->Location = System::Drawing::Point(175, 10);

this->button1->Margin = System::Windows::Forms::Padding(4);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(99, 50);

this->button1->TabIndex = 13;

this->button1->Text = L"<";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &AchivmentsForm::button1\_Click);

//

// button2

//

this->button2->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button2->Location = System::Drawing::Point(785, 10);

this->button2->Margin = System::Windows::Forms::Padding(4);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(99, 50);

this->button2->TabIndex = 15;

this->button2->Text = L">";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &AchivmentsForm::button2\_Click);

//

// button4

//

this->button4->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button4->Location = System::Drawing::Point(892, 10);

this->button4->Margin = System::Windows::Forms::Padding(4);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(99, 50);

this->button4->TabIndex = 22;

this->button4->Text = L"OK";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &AchivmentsForm::button4\_Click);

//

// textBox1

//

this->textBox1->BackColor = System::Drawing::SystemColors::MenuHighlight;

this->textBox1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox1->Location = System::Drawing::Point(23, 10);

this->textBox1->Margin = System::Windows::Forms::Padding(4);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(143, 49);

this->textBox1->TabIndex = 23;

this->textBox1->Text = L" Число";

this->textBox1->TextChanged += gcnew System::EventHandler(this, &AchivmentsForm::textBox1\_TextChanged);

//

// pictureBox1

//

this->pictureBox1->Location = System::Drawing::Point(39, 30);

this->pictureBox1->Margin = System::Windows::Forms::Padding(4);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(133, 123);

this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox1->TabIndex = 24;

this->pictureBox1->TabStop = false;

//

// panel1

//

this->panel1->BackColor = System::Drawing::SystemColors::ControlLight;

this->panel1->Location = System::Drawing::Point(0, 187);

this->panel1->Margin = System::Windows::Forms::Padding(4);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(1057, 5);

this->panel1->TabIndex = 25;

//

// panel2

//

this->panel2->BackColor = System::Drawing::SystemColors::ControlLight;

this->panel2->Location = System::Drawing::Point(39, 596);

this->panel2->Margin = System::Windows::Forms::Padding(4);

this->panel2->Name = L"panel2";

this->panel2->Size = System::Drawing::Size(968, 5);

this->panel2->TabIndex = 26;

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(345, 551);

this->label1->Margin = System::Windows::Forms::Padding(4, 0, 4, 0);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(376, 30);

this->label1->TabIndex = 27;

this->label1->Text = L"Введите дату сдачи экзамена";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->Location = System::Drawing::Point(5, 196);

this->label2->Margin = System::Windows::Forms::Padding(4, 0, 4, 0);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(161, 30);

this->label2->TabIndex = 28;

this->label2->Text = L"Достижения";

//

// label3

//

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label3->Location = System::Drawing::Point(215, 73);

this->label3->Margin = System::Windows::Forms::Padding(4, 0, 4, 0);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(293, 40);

this->label3->TabIndex = 29;

this->label3->Text = L"Имя + Фамилия ";

//

// pictureBox2

//

this->pictureBox2->Location = System::Drawing::Point(963, 15);

this->pictureBox2->Margin = System::Windows::Forms::Padding(4);

this->pictureBox2->Name = L"pictureBox2";

this->pictureBox2->Size = System::Drawing::Size(67, 62);

this->pictureBox2->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox2->TabIndex = 30;

this->pictureBox2->TabStop = false;

this->pictureBox2->Click += gcnew System::EventHandler(this, &AchivmentsForm::pictureBox2\_Click);

//

// pictureBox3

//

this->pictureBox3->BackColor = System::Drawing::Color::Transparent;

this->pictureBox3->Location = System::Drawing::Point(83, 276);

this->pictureBox3->Margin = System::Windows::Forms::Padding(4);

this->pictureBox3->Name = L"pictureBox3";

this->pictureBox3->Size = System::Drawing::Size(107, 98);

this->pictureBox3->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox3->TabIndex = 31;

this->pictureBox3->TabStop = false;

//

// pictureBox4

//

this->pictureBox4->BackColor = System::Drawing::Color::Transparent;

this->pictureBox4->Location = System::Drawing::Point(285, 276);

this->pictureBox4->Margin = System::Windows::Forms::Padding(4);

this->pictureBox4->Name = L"pictureBox4";

this->pictureBox4->Size = System::Drawing::Size(107, 98);

this->pictureBox4->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox4->TabIndex = 32;

this->pictureBox4->TabStop = false;

//

// pictureBox5

//

this->pictureBox5->BackColor = System::Drawing::Color::Transparent;

this->pictureBox5->Location = System::Drawing::Point(476, 276);

this->pictureBox5->Margin = System::Windows::Forms::Padding(4);

this->pictureBox5->Name = L"pictureBox5";

this->pictureBox5->Size = System::Drawing::Size(107, 98);

this->pictureBox5->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox5->TabIndex = 33;

this->pictureBox5->TabStop = false;

//

// pictureBox6

//

this->pictureBox6->BackColor = System::Drawing::Color::Transparent;

this->pictureBox6->Location = System::Drawing::Point(673, 276);

this->pictureBox6->Margin = System::Windows::Forms::Padding(4);

this->pictureBox6->Name = L"pictureBox6";

this->pictureBox6->Size = System::Drawing::Size(107, 98);

this->pictureBox6->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox6->TabIndex = 34;

this->pictureBox6->TabStop = false;

//

// pictureBox7

//

this->pictureBox7->BackColor = System::Drawing::Color::Transparent;

this->pictureBox7->Location = System::Drawing::Point(855, 276);

this->pictureBox7->Margin = System::Windows::Forms::Padding(4);

this->pictureBox7->Name = L"pictureBox7";

this->pictureBox7->Size = System::Drawing::Size(107, 98);

this->pictureBox7->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox7->TabIndex = 35;

this->pictureBox7->TabStop = false;

//

// pictureBox8

//

this->pictureBox8->Location = System::Drawing::Point(68, 405);

this->pictureBox8->Margin = System::Windows::Forms::Padding(4);

this->pictureBox8->Name = L"pictureBox8";

this->pictureBox8->Size = System::Drawing::Size(40, 37);

this->pictureBox8->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox8->TabIndex = 36;

this->pictureBox8->TabStop = false;

//

// pictureBox9

//

this->pictureBox9->Location = System::Drawing::Point(116, 417);

this->pictureBox9->Margin = System::Windows::Forms::Padding(4);

this->pictureBox9->Name = L"pictureBox9";

this->pictureBox9->Size = System::Drawing::Size(40, 37);

this->pictureBox9->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox9->TabIndex = 37;

this->pictureBox9->TabStop = false;

//

// pictureBox10

//

this->pictureBox10->Location = System::Drawing::Point(165, 405);

this->pictureBox10->Margin = System::Windows::Forms::Padding(4);

this->pictureBox10->Name = L"pictureBox10";

this->pictureBox10->Size = System::Drawing::Size(40, 37);

this->pictureBox10->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox10->TabIndex = 38;

this->pictureBox10->TabStop = false;

//

// pictureBox12

//

this->pictureBox12->Location = System::Drawing::Point(116, 417);

this->pictureBox12->Margin = System::Windows::Forms::Padding(4);

this->pictureBox12->Name = L"pictureBox12";

this->pictureBox12->Size = System::Drawing::Size(40, 37);

this->pictureBox12->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox12->TabIndex = 37;

this->pictureBox12->TabStop = false;

//

// pictureBox13

//

this->pictureBox13->Location = System::Drawing::Point(165, 405);

this->pictureBox13->Margin = System::Windows::Forms::Padding(4);

this->pictureBox13->Name = L"pictureBox13";

this->pictureBox13->Size = System::Drawing::Size(40, 37);

this->pictureBox13->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox13->TabIndex = 38;

this->pictureBox13->TabStop = false;

//

// pictureBox14

//

this->pictureBox14->Location = System::Drawing::Point(365, 405);

this->pictureBox14->Margin = System::Windows::Forms::Padding(4);

this->pictureBox14->Name = L"pictureBox14";

this->pictureBox14->Size = System::Drawing::Size(40, 37);

this->pictureBox14->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox14->TabIndex = 41;

this->pictureBox14->TabStop = false;

//

// pictureBox15

//

this->pictureBox15->Location = System::Drawing::Point(316, 417);

this->pictureBox15->Margin = System::Windows::Forms::Padding(4);

this->pictureBox15->Name = L"pictureBox15";

this->pictureBox15->Size = System::Drawing::Size(40, 37);

this->pictureBox15->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox15->TabIndex = 40;

this->pictureBox15->TabStop = false;

//

// pictureBox16

//

this->pictureBox16->Location = System::Drawing::Point(267, 405);

this->pictureBox16->Margin = System::Windows::Forms::Padding(4);

this->pictureBox16->Name = L"pictureBox16";

this->pictureBox16->Size = System::Drawing::Size(40, 37);

this->pictureBox16->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox16->TabIndex = 39;

this->pictureBox16->TabStop = false;

//

// pictureBox17

//

this->pictureBox17->Location = System::Drawing::Point(561, 405);

this->pictureBox17->Margin = System::Windows::Forms::Padding(4);

this->pictureBox17->Name = L"pictureBox17";

this->pictureBox17->Size = System::Drawing::Size(40, 37);

this->pictureBox17->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox17->TabIndex = 44;

this->pictureBox17->TabStop = false;

//

// pictureBox18

//

this->pictureBox18->Location = System::Drawing::Point(512, 417);

this->pictureBox18->Margin = System::Windows::Forms::Padding(4);

this->pictureBox18->Name = L"pictureBox18";

this->pictureBox18->Size = System::Drawing::Size(40, 37);

this->pictureBox18->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox18->TabIndex = 43;

this->pictureBox18->TabStop = false;

//

// pictureBox19

//

this->pictureBox19->Location = System::Drawing::Point(463, 405);

this->pictureBox19->Margin = System::Windows::Forms::Padding(4);

this->pictureBox19->Name = L"pictureBox19";

this->pictureBox19->Size = System::Drawing::Size(40, 37);

this->pictureBox19->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox19->TabIndex = 42;

this->pictureBox19->TabStop = false;

//

// pictureBox20

//

this->pictureBox20->Location = System::Drawing::Point(751, 405);

this->pictureBox20->Margin = System::Windows::Forms::Padding(4);

this->pictureBox20->Name = L"pictureBox20";

this->pictureBox20->Size = System::Drawing::Size(40, 37);

this->pictureBox20->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox20->TabIndex = 47;

this->pictureBox20->TabStop = false;

//

// pictureBox21

//

this->pictureBox21->Location = System::Drawing::Point(701, 417);

this->pictureBox21->Margin = System::Windows::Forms::Padding(4);

this->pictureBox21->Name = L"pictureBox21";

this->pictureBox21->Size = System::Drawing::Size(40, 37);

this->pictureBox21->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox21->TabIndex = 46;

this->pictureBox21->TabStop = false;

//

// pictureBox22

//

this->pictureBox22->Location = System::Drawing::Point(652, 405);

this->pictureBox22->Margin = System::Windows::Forms::Padding(4);

this->pictureBox22->Name = L"pictureBox22";

this->pictureBox22->Size = System::Drawing::Size(40, 37);

this->pictureBox22->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox22->TabIndex = 45;

this->pictureBox22->TabStop = false;

//

// pictureBox23

//

this->pictureBox23->Location = System::Drawing::Point(939, 405);

this->pictureBox23->Margin = System::Windows::Forms::Padding(4);

this->pictureBox23->Name = L"pictureBox23";

this->pictureBox23->Size = System::Drawing::Size(40, 37);

this->pictureBox23->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox23->TabIndex = 50;

this->pictureBox23->TabStop = false;

//

// pictureBox24

//

this->pictureBox24->Location = System::Drawing::Point(889, 417);

this->pictureBox24->Margin = System::Windows::Forms::Padding(4);

this->pictureBox24->Name = L"pictureBox24";

this->pictureBox24->Size = System::Drawing::Size(40, 37);

this->pictureBox24->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox24->TabIndex = 49;

this->pictureBox24->TabStop = false;

//

// pictureBox25

//

this->pictureBox25->Location = System::Drawing::Point(840, 405);

this->pictureBox25->Margin = System::Windows::Forms::Padding(4);

this->pictureBox25->Name = L"pictureBox25";

this->pictureBox25->Size = System::Drawing::Size(40, 37);

this->pictureBox25->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox25->TabIndex = 48;

this->pictureBox25->TabStop = false;

//

// panel3

//

this->panel3->BackColor = System::Drawing::SystemColors::ActiveBorder;

this->panel3->Location = System::Drawing::Point(68, 263);

this->panel3->Margin = System::Windows::Forms::Padding(4);

this->panel3->Name = L"panel3";

this->panel3->Size = System::Drawing::Size(133, 123);

this->panel3->TabIndex = 51;

//

// panel4

//

this->panel4->BackColor = System::Drawing::SystemColors::ActiveBorder;

this->panel4->Location = System::Drawing::Point(272, 263);

this->panel4->Margin = System::Windows::Forms::Padding(4);

this->panel4->Name = L"panel4";

this->panel4->Size = System::Drawing::Size(133, 123);

this->panel4->TabIndex = 52;

//

// panel5

//

this->panel5->BackColor = System::Drawing::SystemColors::ActiveBorder;

this->panel5->Location = System::Drawing::Point(463, 263);

this->panel5->Margin = System::Windows::Forms::Padding(4);

this->panel5->Name = L"panel5";

this->panel5->Size = System::Drawing::Size(133, 123);

this->panel5->TabIndex = 52;

//

// panel6

//

this->panel6->BackColor = System::Drawing::SystemColors::ActiveBorder;

this->panel6->Location = System::Drawing::Point(660, 263);

this->panel6->Margin = System::Windows::Forms::Padding(4);

this->panel6->Name = L"panel6";

this->panel6->Size = System::Drawing::Size(133, 123);

this->panel6->TabIndex = 52;

//

// panel7

//

this->panel7->BackColor = System::Drawing::SystemColors::ActiveBorder;

this->panel7->Location = System::Drawing::Point(841, 263);

this->panel7->Margin = System::Windows::Forms::Padding(4);

this->panel7->Name = L"panel7";

this->panel7->Size = System::Drawing::Size(133, 123);

this->panel7->TabIndex = 52;

//

// toolTip1

//

this->toolTip1->Tag = L"pictureBox3";

this->toolTip1->ToolTipTitle = L"Мям";

this->toolTip1->Popup += gcnew System::Windows::Forms::PopupEventHandler(this, &AchivmentsForm::toolTip1\_Popup);

//

// timer1

//

this->timer1->Enabled = true;

this->timer1->Interval = 10;

this->timer1->Tick += gcnew System::EventHandler(this, &AchivmentsForm::timer1\_Tick);

//

// panel8

//

this->panel8->Controls->Add(this->label5);

this->panel8->Controls->Add(this->textBox1);

this->panel8->Controls->Add(this->label4);

this->panel8->Controls->Add(this->textBox2);

this->panel8->Controls->Add(this->button1);

this->panel8->Controls->Add(this->button2);

this->panel8->Controls->Add(this->button4);

this->panel8->Location = System::Drawing::Point(16, 615);

this->panel8->Margin = System::Windows::Forms::Padding(4);

this->panel8->Name = L"panel8";

this->panel8->Size = System::Drawing::Size(1021, 66);

this->panel8->TabIndex = 53;

//

// label5

//

this->label5->AutoSize = true;

this->label5->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label5->Location = System::Drawing::Point(741, 17);

this->label5->Margin = System::Windows::Forms::Padding(4, 0, 4, 0);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(232, 37);

this->label5->TabIndex = 55;

this->label5->Text = L"Изменить дату";

this->label5->Click += gcnew System::EventHandler(this, &AchivmentsForm::label5\_Click);

//

// label4

//

this->label4->AutoSize = true;

this->label4->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label4->Location = System::Drawing::Point(16, 17);

this->label4->Margin = System::Windows::Forms::Padding(4, 0, 4, 0);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(462, 37);

this->label4->TabIndex = 54;

this->label4->Text = L"Введите дату сдачи экзамена";

//

// panel9

//

this->panel9->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel9->Controls->Add(this->panel8);

this->panel9->Controls->Add(this->pictureBox23);

this->panel9->Controls->Add(this->pictureBox24);

this->panel9->Controls->Add(this->pictureBox25);

this->panel9->Controls->Add(this->pictureBox20);

this->panel9->Controls->Add(this->pictureBox21);

this->panel9->Controls->Add(this->pictureBox22);

this->panel9->Controls->Add(this->pictureBox17);

this->panel9->Controls->Add(this->pictureBox18);

this->panel9->Controls->Add(this->pictureBox19);

this->panel9->Controls->Add(this->pictureBox14);

this->panel9->Controls->Add(this->pictureBox15);

this->panel9->Controls->Add(this->pictureBox16);

this->panel9->Controls->Add(this->pictureBox13);

this->panel9->Controls->Add(this->pictureBox10);

this->panel9->Controls->Add(this->pictureBox12);

this->panel9->Controls->Add(this->pictureBox9);

this->panel9->Controls->Add(this->pictureBox8);

this->panel9->Controls->Add(this->pictureBox7);

this->panel9->Controls->Add(this->pictureBox6);

this->panel9->Controls->Add(this->pictureBox5);

this->panel9->Controls->Add(this->pictureBox4);

this->panel9->Controls->Add(this->pictureBox3);

this->panel9->Controls->Add(this->pictureBox2);

this->panel9->Controls->Add(this->label3);

this->panel9->Controls->Add(this->label2);

this->panel9->Controls->Add(this->label1);

this->panel9->Controls->Add(this->panel2);

this->panel9->Controls->Add(this->panel1);

this->panel9->Controls->Add(this->pictureBox1);

this->panel9->Controls->Add(this->panel3);

this->panel9->Controls->Add(this->panel4);

this->panel9->Controls->Add(this->panel5);

this->panel9->Controls->Add(this->panel6);

this->panel9->Controls->Add(this->panel7);

this->panel9->Location = System::Drawing::Point(0, 0);

this->panel9->Margin = System::Windows::Forms::Padding(4);

this->panel9->Name = L"panel9";

this->panel9->Size = System::Drawing::Size(1045, 689);

this->panel9->TabIndex = 54;

this->panel9->Paint += gcnew System::Windows::Forms::PaintEventHandler(this, &AchivmentsForm::panel9\_Paint);

//

// ComfortPlus

//

this->ComfortPlus->Enabled = true;

this->ComfortPlus->Tick += gcnew System::EventHandler(this, &AchivmentsForm::ComfortPlus\_Tick);

//

// AchivmentsForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(1045, 690);

this->ControlBox = false;

this->Controls->Add(this->panel9);

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"AchivmentsForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Достижения";

this->Load += gcnew System::EventHandler(this, &AchivmentsForm::AchivmentsForm\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &AchivmentsForm::AchivmentsForm\_VisibleChanged);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox2))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox3))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox4))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox5))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox6))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox7))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox8))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox9))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox10))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox12))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox13))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox14))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox15))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox16))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox17))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox18))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox19))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox20))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox21))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox22))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox23))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox24))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox25))->EndInit();

this->panel8->ResumeLayout(false);

this->panel8->PerformLayout();

this->panel9->ResumeLayout(false);

this->panel9->PerformLayout();

this->ResumeLayout(false);

}

#pragma endregion

int sec;

int SelectedDay;

int SelectedMonth;

int Achivment\_1 = Achivments::Ac\_sel("0", ConvertToString(Login\_Text));

int Achivment\_2 = Achivments::Ac\_sel("1", ConvertToString(Login\_Text));

int Achivment\_3 = Achivments::Ac\_sel("2", ConvertToString(Login\_Text));

int Achivment\_4 = Achivments::Ac\_sel("3", ConvertToString(Login\_Text));

int Achivment\_5 = Achivments::Ac\_sel("4", ConvertToString(Login\_Text));

int MonthCount = 1;

int DayCount = 1;

void MonthChanger()

{

if (MonthCount == 1)

textBox2->Text = "Январь";

if (MonthCount == 2)

textBox2->Text = "Февраль";

if (MonthCount == 3)

textBox2->Text = "Март";

if (MonthCount == 4)

textBox2->Text = "Апрель";

if (MonthCount == 5)

textBox2->Text = "Май";

if (MonthCount == 6)

textBox2->Text = "Июнь";

if (MonthCount == 7)

textBox2->Text = "Июль";

if (MonthCount == 8)

textBox2->Text = "Август";

if (MonthCount == 9)

textBox2->Text = "Сентябрь";

if (MonthCount == 10)

textBox2->Text = "Октябрь";

if (MonthCount == 11)

textBox2->Text = "Ноябрь";

if (MonthCount == 12)

textBox2->Text = "Декабрь";

if (MonthCount == 13)

{

textBox2->Text = "Январь";

MonthCount = 1;

}

if (MonthCount == 0)

{

textBox2->Text = "Декабрь";

MonthCount = 12;

}

}

void Achivka(PictureBox^PB1, PictureBox^PB2, PictureBox^PB3, int Index)

{

if (Index == 1)

{

PB1->Image = Image::FromFile(Application::StartupPath + "\\UI\\Star.png");

}

if (Index == 2)

{

PB1->Image = Image::FromFile(Application::StartupPath + "\\UI\\Star.png");

PB2->Image = Image::FromFile(Application::StartupPath + "\\UI\\Star.png");

}

if (Index == 3)

{

PB1->Image = Image::FromFile(Application::StartupPath + "\\UI\\Star.png");

PB2->Image = Image::FromFile(Application::StartupPath + "\\UI\\Star.png");

PB3->Image = Image::FromFile(Application::StartupPath + "\\UI\\Star.png");

}

}

private: System::Void label12\_Click(System::Object^ sender, System::EventArgs^ e)

{

MessageBox::Show("Вы молодец");

}

void Select\_Name(String ^ &ReturnText) {

auto p = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application::StartupPath + "\\resource\\Database1.mdb");

p->Open();

auto v = gcnew OleDbCommand("SELECT \* FROM [MAIN]", p);

v->CommandText = "SELECT [Имя] FROM [MAIN] WHERE [Login] = '" + ConvertToString(Login\_Text) + "'";

auto s = gcnew OleDbCommand("SELECT \* FROM [MAIN]", p);

s->CommandText = "SELECT [Фамилия] FROM [MAIN] WHERE [Login] = '" + ConvertToString(Login\_Text) + "'";

ReturnText = Convert::ToString(v->ExecuteScalar()) + " " + Convert::ToString(s->ExecuteScalar());

p->Close();

}

private: System::Void AchivmentsForm\_Load(System::Object^ sender, System::EventArgs^ e) {

String ^ Name;

Select\_Name(Name);

label3->Text = Name;

label4->Visible = 0;

label5->Visible = 0;

pictureBox3->Image = Image::FromFile(Application::StartupPath + "\\UI\\Pereskaz.png");

pictureBox4->Image = Image::FromFile(Application::StartupPath + "\\UI\\Writing.png");

pictureBox5->Image = Image::FromFile(Application::StartupPath + "\\UI\\Theory.png");

pictureBox6->Image = Image::FromFile(Application::StartupPath + "\\UI\\Testing.png");

pictureBox7->Image = Image::FromFile(Application::StartupPath + "\\UI\\Calendar.png");

if (Achivment\_1 > 0)

{

panel3->BackColor = System::Drawing::Color::MediumOrchid;

}

if (Achivment\_2 > 0)

{

panel4->BackColor = System::Drawing::Color::Red;

}

if (Achivment\_3 > 0)

{

panel5->BackColor = System::Drawing::Color::DarkOrange;

}

if (Achivment\_4 > 0)

{

panel6->BackColor = System::Drawing::Color::DodgerBlue;

}

if (Achivment\_5 > 0)

{

panel7->BackColor = System::Drawing::Color::LimeGreen;

}

Achivka(pictureBox8, pictureBox12, pictureBox13, Achivment\_1);

Achivka(pictureBox16, pictureBox15, pictureBox14, Achivment\_2);

Achivka(pictureBox19, pictureBox18, pictureBox17, Achivment\_3);

Achivka(pictureBox22, pictureBox21, pictureBox20, Achivment\_4);

Achivka(pictureBox25, pictureBox24, pictureBox23, Achivment\_5);

pictureBox1->Image = Image::FromFile(Application::StartupPath + "\\UI\\User.png");

pictureBox2->Image = Image::FromFile(Application::StartupPath + "\\UI\\Arrow.png");

MonthChanger();

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e)

{

int Checket = (Convert::ToInt16(textBox1->Text));

if ((Checket == 1) || (Checket == 2) || (Checket == 3) || (Checket == 4) || (Checket == 5) || (Checket == 6) || (Checket == 7) || (Checket == 8 )|| (Checket == 9) || (Checket == 10)|| (Checket == 11)|| (Checket == 12) || (Checket == 13) || (Checket == 14) || (Checket == 15) || (Checket == 16) || (Checket == 17) || (Checket == 18) || (Checket == 19) || (Checket == 20) || (Checket == 21) || (Checket == 22) || (Checket == 23) || (Checket == 24) || (Checket == 25) || (Checket == 26) || (Checket == 27) || (Checket == 28) || (Checket == 29) || (Checket == 30) || (Checket == 31))

{

label4->Visible = 1;

label5->Visible = 1;

if (textBox1->Text == "") {

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

label4->Visible = 0;

label5->Visible = 0;

}

else {

int TheDay = Convert::ToInt32(textBox1->Text);

if (TheDay > 31 && MonthCount == 1)

{

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

label4->Visible = 0;

label5->Visible = 0;

}

else if (TheDay > 28 && MonthCount == 2)

{

textBox1->Text = "";

label4->Visible = 0;

label5->Visible = 0;

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 31 && MonthCount == 3)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 30 && MonthCount == 4)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 31 && MonthCount == 5)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 30 && MonthCount == 6)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 31 && MonthCount == 7)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 31 && MonthCount == 8)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 30 && MonthCount == 9)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 31 && MonthCount == 10)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 30 && MonthCount == 11)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else if (TheDay > 31 && MonthCount == 21)

{

label4->Visible = 0;

label5->Visible = 0;

textBox1->Text = "";

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

else {

SelectedDay = TheDay;

SelectedMonth = MonthCount;

textBox1->Visible = 0;

button1->Visible = 0;

textBox2->Visible = 0;

button2->Visible = 0;

button4->Visible = 0;

label4->Text = "Вы сдаете экзамен " + Convert::ToString(SelectedDay) + " . " + Convert::ToString(SelectedMonth);

CalendarGlobal[0] = SelectedDay;

CalendarGlobal[1] = SelectedMonth;

}

}

}

else

{

label4->Visible = 0;

label5->Visible = 0;

MessageBox::Show("Пожалуйста, проверьте корректность введенных данных", "Проверьте данные", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

}

private: System::Void pictureBox2\_Click(System::Object^ sender, System::EventArgs^ e)

{

Owner->Show();

this->Hide();

}

private: System::Void toolTip1\_Popup(System::Object^ sender, System::Windows::Forms::PopupEventArgs^ e)

{

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

MonthCount = MonthCount - 1;

MonthChanger();

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e)

{

MonthCount++;

MonthChanger();

}

private: System::Void timer1\_Tick(System::Object^ sender, System::EventArgs^ e)

{

}

private: System::Void label5\_Click(System::Object^ sender, System::EventArgs^ e)

{

textBox1->Visible = 1;

button1->Visible = 1;

textBox2->Visible = 1;

button2->Visible = 1;

button4->Visible = 1;

label4->Visible = 0;

label5->Visible = 0;

}

private: System::Void AchivmentsForm\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

private: System::Void textBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

if (textBox1->Text == " ") {

textBox1->Text = "";

}

}

private: System::Void panel9\_Paint(System::Object^ sender, System::Windows::Forms::PaintEventArgs^ e) {

}

private: System::Void ComfortPlus\_Tick(System::Object^ sender, System::EventArgs^ e)

{

sec++;

if (sec == 30)

{

textBox1->Text = "";

textBox1->BackColor = System::Drawing::Color::White;

}

}

};

}

**AllTests.h**

#pragma once

#include "TestsHeaders.h"

#include "VAR.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace std;

using namespace BrokenPiano;

using namespace Obj\_Space;

/// <summary>

/// Сводка для AllTests

/// </summary>

public ref class AllTests : public System::Windows::Forms::Form

{

public:

AllTests(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~AllTests()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Panel^ panel1;

private: System::Windows::Forms::TrackBar^ trackBar1;

private: System::Windows::Forms::Button^ button11;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Button^ button10;

private: System::Windows::Forms::Button^ button9;

private: System::Windows::Forms::Button^ button8;

private: System::Windows::Forms::Button^ button7;

private: System::Windows::Forms::Button^ button6;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button12;

private: System::Windows::Forms::Panel^ panel2;

private: System::Windows::Forms::Panel^ panel3;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::Button^ button13;

private: System::Windows::Forms::TrackBar^ trackBar2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::Button^ button14;

private: System::Windows::Forms::GroupBox^ groupBox1;

private: System::Windows::Forms::RadioButton^ radioButton3;

private: System::Windows::Forms::RadioButton^ radioButton4;

private: System::Windows::Forms::RadioButton^ radioButton1;

private: System::Windows::Forms::RadioButton^ radioButton2;

private: System::Windows::Forms::TrackBar^ trackBar3;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::Panel^ panel7;

private: System::Windows::Forms::Button^ button22;

private: System::Windows::Forms::DataGridView^ dataGridView1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->trackBar1 = (gcnew System::Windows::Forms::TrackBar());

this->button11 = (gcnew System::Windows::Forms::Button());

this->label1 = (gcnew System::Windows::Forms::Label());

this->button10 = (gcnew System::Windows::Forms::Button());

this->button9 = (gcnew System::Windows::Forms::Button());

this->button8 = (gcnew System::Windows::Forms::Button());

this->button7 = (gcnew System::Windows::Forms::Button());

this->button6 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button12 = (gcnew System::Windows::Forms::Button());

this->panel2 = (gcnew System::Windows::Forms::Panel());

this->label3 = (gcnew System::Windows::Forms::Label());

this->button13 = (gcnew System::Windows::Forms::Button());

this->trackBar2 = (gcnew System::Windows::Forms::TrackBar());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->label2 = (gcnew System::Windows::Forms::Label());

this->panel3 = (gcnew System::Windows::Forms::Panel());

this->label5 = (gcnew System::Windows::Forms::Label());

this->trackBar3 = (gcnew System::Windows::Forms::TrackBar());

this->button14 = (gcnew System::Windows::Forms::Button());

this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());

this->radioButton3 = (gcnew System::Windows::Forms::RadioButton());

this->radioButton4 = (gcnew System::Windows::Forms::RadioButton());

this->radioButton1 = (gcnew System::Windows::Forms::RadioButton());

this->radioButton2 = (gcnew System::Windows::Forms::RadioButton());

this->label4 = (gcnew System::Windows::Forms::Label());

this->panel7 = (gcnew System::Windows::Forms::Panel());

this->button22 = (gcnew System::Windows::Forms::Button());

this->dataGridView1 = (gcnew System::Windows::Forms::DataGridView());

this->panel1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->trackBar1))->BeginInit();

this->panel2->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->trackBar2))->BeginInit();

this->panel3->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->trackBar3))->BeginInit();

this->groupBox1->SuspendLayout();

this->panel7->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->BeginInit();

this->SuspendLayout();

//

// button1

//

this->button1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button1->Location = System::Drawing::Point(577, 518);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(195, 31);

this->button1->TabIndex = 0;

this->button1->Text = L"Завершить тест досрочно";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &AllTests::button1\_Click);

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->trackBar1);

this->panel1->Controls->Add(this->button11);

this->panel1->Controls->Add(this->label1);

this->panel1->Controls->Add(this->button10);

this->panel1->Controls->Add(this->button9);

this->panel1->Controls->Add(this->button8);

this->panel1->Controls->Add(this->button7);

this->panel1->Controls->Add(this->button6);

this->panel1->Controls->Add(this->button5);

this->panel1->Controls->Add(this->button4);

this->panel1->Controls->Add(this->button3);

this->panel1->Controls->Add(this->button2);

this->panel1->Location = System::Drawing::Point(12, 12);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(760, 500);

this->panel1->TabIndex = 1;

this->panel1->VisibleChanged += gcnew System::EventHandler(this, &AllTests::panel1\_VisibleChanged);

//

// trackBar1

//

this->trackBar1->LargeChange = 0;

this->trackBar1->Location = System::Drawing::Point(93, 22);

this->trackBar1->Maximum = 4;

this->trackBar1->Name = L"trackBar1";

this->trackBar1->Size = System::Drawing::Size(581, 45);

this->trackBar1->TabIndex = 0;

this->trackBar1->TabStop = false;

this->trackBar1->Scroll += gcnew System::EventHandler(this, &AllTests::trackBar1\_Scroll);

//

// button11

//

this->button11->Location = System::Drawing::Point(301, 420);

this->button11->Name = L"button11";

this->button11->Size = System::Drawing::Size(166, 48);

this->button11->TabIndex = 10;

this->button11->Text = L"Ответить";

this->button11->UseVisualStyleBackColor = true;

this->button11->Click += gcnew System::EventHandler(this, &AllTests::button11\_Click);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(312, 106);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(146, 22);

this->label1->TabIndex = 9;

this->label1->Text = L"Текст вопроса";

this->label1->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;

//

// button10

//

this->button10->Location = System::Drawing::Point(508, 320);

this->button10->Name = L"button10";

this->button10->Size = System::Drawing::Size(166, 48);

this->button10->TabIndex = 8;

this->button10->Text = L"button10";

this->button10->UseVisualStyleBackColor = true;

this->button10->Click += gcnew System::EventHandler(this, &AllTests::button10\_Click);

//

// button9

//

this->button9->Location = System::Drawing::Point(301, 320);

this->button9->Name = L"button9";

this->button9->Size = System::Drawing::Size(166, 48);

this->button9->TabIndex = 7;

this->button9->Text = L"button9";

this->button9->UseVisualStyleBackColor = true;

this->button9->Click += gcnew System::EventHandler(this, &AllTests::button9\_Click);

//

// button8

//

this->button8->Location = System::Drawing::Point(93, 320);

this->button8->Name = L"button8";

this->button8->Size = System::Drawing::Size(166, 48);

this->button8->TabIndex = 6;

this->button8->Text = L"button8";

this->button8->UseVisualStyleBackColor = true;

this->button8->Click += gcnew System::EventHandler(this, &AllTests::button8\_Click);

//

// button7

//

this->button7->Location = System::Drawing::Point(508, 266);

this->button7->Name = L"button7";

this->button7->Size = System::Drawing::Size(166, 48);

this->button7->TabIndex = 5;

this->button7->Text = L"button7";

this->button7->UseVisualStyleBackColor = true;

this->button7->Click += gcnew System::EventHandler(this, &AllTests::button7\_Click);

//

// button6

//

this->button6->Location = System::Drawing::Point(301, 266);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(166, 48);

this->button6->TabIndex = 4;

this->button6->Text = L"button6";

this->button6->UseVisualStyleBackColor = true;

this->button6->Click += gcnew System::EventHandler(this, &AllTests::button6\_Click);

//

// button5

//

this->button5->Location = System::Drawing::Point(93, 266);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(166, 48);

this->button5->TabIndex = 3;

this->button5->Text = L"button5";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &AllTests::button5\_Click);

//

// button4

//

this->button4->Location = System::Drawing::Point(508, 161);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(166, 48);

this->button4->TabIndex = 2;

this->button4->Text = L"button4";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &AllTests::button4\_Click);

//

// button3

//

this->button3->Location = System::Drawing::Point(301, 161);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(166, 48);

this->button3->TabIndex = 1;

this->button3->Text = L"button3";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &AllTests::button3\_Click);

//

// button2

//

this->button2->Location = System::Drawing::Point(93, 161);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(166, 48);

this->button2->TabIndex = 0;

this->button2->Text = L"button2";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &AllTests::button2\_Click);

//

// button12

//

this->button12->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button12->Location = System::Drawing::Point(295, 518);

this->button12->Name = L"button12";

this->button12->Size = System::Drawing::Size(195, 31);

this->button12->TabIndex = 2;

this->button12->Text = L"Следующий вопрос";

this->button12->UseVisualStyleBackColor = true;

this->button12->Click += gcnew System::EventHandler(this, &AllTests::button12\_Click);

//

// panel2

//

this->panel2->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel2->Controls->Add(this->label3);

this->panel2->Controls->Add(this->button13);

this->panel2->Controls->Add(this->trackBar2);

this->panel2->Controls->Add(this->textBox1);

this->panel2->Controls->Add(this->label2);

this->panel2->Location = System::Drawing::Point(12, 12);

this->panel2->Name = L"panel2";

this->panel2->Size = System::Drawing::Size(760, 500);

this->panel2->TabIndex = 3;

this->panel2->VisibleChanged += gcnew System::EventHandler(this, &AllTests::panel2\_VisibleChanged);

//

// label3

//

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label3->Location = System::Drawing::Point(11, 302);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(65, 22);

this->label3->TabIndex = 4;

this->label3->Text = L"label3";

//

// button13

//

this->button13->Location = System::Drawing::Point(301, 375);

this->button13->Name = L"button13";

this->button13->Size = System::Drawing::Size(166, 39);

this->button13->TabIndex = 3;

this->button13->Text = L"Ответить";

this->button13->UseVisualStyleBackColor = true;

this->button13->Click += gcnew System::EventHandler(this, &AllTests::button13\_Click);

//

// trackBar2

//

this->trackBar2->Location = System::Drawing::Point(15, 22);

this->trackBar2->Maximum = 4;

this->trackBar2->Name = L"trackBar2";

this->trackBar2->Size = System::Drawing::Size(720, 45);

this->trackBar2->TabIndex = 0;

this->trackBar2->TabStop = false;

this->trackBar2->Scroll += gcnew System::EventHandler(this, &AllTests::trackBar2\_Scroll);

//

// textBox1

//

this->textBox1->BorderStyle = System::Windows::Forms::BorderStyle::FixedSingle;

this->textBox1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox1->Location = System::Drawing::Point(15, 300);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(720, 31);

this->textBox1->TabIndex = 2;

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->Location = System::Drawing::Point(11, 92);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(146, 22);

this->label2->TabIndex = 1;

this->label2->Text = L"Текст вопроса";

//

// panel3

//

this->panel3->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel3->Controls->Add(this->label5);

this->panel3->Controls->Add(this->trackBar3);

this->panel3->Controls->Add(this->button14);

this->panel3->Controls->Add(this->groupBox1);

this->panel3->Controls->Add(this->label4);

this->panel3->Location = System::Drawing::Point(12, 12);

this->panel3->Name = L"panel3";

this->panel3->Size = System::Drawing::Size(760, 500);

this->panel3->TabIndex = 4;

this->panel3->VisibleChanged += gcnew System::EventHandler(this, &AllTests::panel3\_VisibleChanged);

//

// label5

//

this->label5->AutoSize = true;

this->label5->Location = System::Drawing::Point(33, 237);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(43, 17);

this->label5->TabIndex = 7;

this->label5->Text = L"Ответ";

//

// trackBar3

//

this->trackBar3->LargeChange = 0;

this->trackBar3->Location = System::Drawing::Point(24, 22);

this->trackBar3->Maximum = 4;

this->trackBar3->Name = L"trackBar3";

this->trackBar3->Size = System::Drawing::Size(711, 45);

this->trackBar3->TabIndex = 0;

this->trackBar3->Scroll += gcnew System::EventHandler(this, &AllTests::trackBar3\_Scroll);

//

// button14

//

this->button14->Location = System::Drawing::Point(283, 421);

this->button14->Name = L"button14";

this->button14->Size = System::Drawing::Size(195, 47);

this->button14->TabIndex = 6;

this->button14->Text = L"Ответить";

this->button14->UseVisualStyleBackColor = true;

this->button14->Click += gcnew System::EventHandler(this, &AllTests::button14\_Click);

//

// groupBox1

//

this->groupBox1->Controls->Add(this->radioButton3);

this->groupBox1->Controls->Add(this->radioButton4);

this->groupBox1->Controls->Add(this->radioButton1);

this->groupBox1->Controls->Add(this->radioButton2);

this->groupBox1->Location = System::Drawing::Point(24, 266);

this->groupBox1->Name = L"groupBox1";

this->groupBox1->Size = System::Drawing::Size(711, 146);

this->groupBox1->TabIndex = 5;

this->groupBox1->TabStop = false;

//

// radioButton3

//

this->radioButton3->AutoSize = true;

this->radioButton3->Location = System::Drawing::Point(15, 76);

this->radioButton3->Name = L"radioButton3";

this->radioButton3->Size = System::Drawing::Size(109, 21);

this->radioButton3->TabIndex = 3;

this->radioButton3->TabStop = true;

this->radioButton3->Text = L"radioButton3";

this->radioButton3->UseVisualStyleBackColor = true;

//

// radioButton4

//

this->radioButton4->AutoSize = true;

this->radioButton4->Location = System::Drawing::Point(15, 103);

this->radioButton4->Name = L"radioButton4";

this->radioButton4->Size = System::Drawing::Size(109, 21);

this->radioButton4->TabIndex = 4;

this->radioButton4->TabStop = true;

this->radioButton4->Text = L"radioButton4";

this->radioButton4->UseVisualStyleBackColor = true;

//

// radioButton1

//

this->radioButton1->AutoSize = true;

this->radioButton1->Location = System::Drawing::Point(15, 22);

this->radioButton1->Name = L"radioButton1";

this->radioButton1->Size = System::Drawing::Size(109, 21);

this->radioButton1->TabIndex = 1;

this->radioButton1->TabStop = true;

this->radioButton1->Text = L"radioButton1";

this->radioButton1->UseVisualStyleBackColor = true;

//

// radioButton2

//

this->radioButton2->AutoSize = true;

this->radioButton2->Location = System::Drawing::Point(15, 49);

this->radioButton2->Name = L"radioButton2";

this->radioButton2->Size = System::Drawing::Size(109, 21);

this->radioButton2->TabIndex = 2;

this->radioButton2->TabStop = true;

this->radioButton2->Text = L"radioButton2";

this->radioButton2->UseVisualStyleBackColor = true;

//

// label4

//

this->label4->AutoSize = true;

this->label4->Location = System::Drawing::Point(93, 74);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(108, 17);

this->label4->TabIndex = 0;

this->label4->Text = L"Тип вопросов 3";

//

// panel7

//

this->panel7->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel7->Controls->Add(this->button22);

this->panel7->Controls->Add(this->dataGridView1);

this->panel7->Location = System::Drawing::Point(12, 12);

this->panel7->Name = L"panel7";

this->panel7->Size = System::Drawing::Size(760, 537);

this->panel7->TabIndex = 8;

//

// button22

//

this->button22->Location = System::Drawing::Point(15, 485);

this->button22->Name = L"button22";

this->button22->Size = System::Drawing::Size(197, 38);

this->button22->TabIndex = 1;

this->button22->Text = L"Вернуться";

this->button22->UseVisualStyleBackColor = true;

this->button22->Click += gcnew System::EventHandler(this, &AllTests::button22\_Click);

//

// dataGridView1

//

this->dataGridView1->BackgroundColor = System::Drawing::SystemColors::Control;

this->dataGridView1->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView1->Location = System::Drawing::Point(15, 20);

this->dataGridView1->Name = L"dataGridView1";

this->dataGridView1->Size = System::Drawing::Size(720, 189);

this->dataGridView1->TabIndex = 0;

//

// AllTests

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 17);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(784, 561);

this->ControlBox = false;

this->Controls->Add(this->button12);

this->Controls->Add(this->button1);

this->Controls->Add(this->panel2);

this->Controls->Add(this->panel1);

this->Controls->Add(this->panel7);

this->Controls->Add(this->panel3);

this->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"AllTests";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Тестирование";

this->Load += gcnew System::EventHandler(this, &AllTests::AllTests\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &AllTests::AllTests\_VisibleChanged);

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->trackBar1))->EndInit();

this->panel2->ResumeLayout(false);

this->panel2->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->trackBar2))->EndInit();

this->panel3->ResumeLayout(false);

this->panel3->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->trackBar3))->EndInit();

this->groupBox1->ResumeLayout(false);

this->groupBox1->PerformLayout();

this->panel7->ResumeLayout(false);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->EndInit();

this->ResumeLayout(false);

}

#pragma endregion

int \*now\_test = new int[2];

int \*ans\_kol = new int[1];

bool \*now\_qst = new bool[4];

bool \*qst = new bool[5];

int \*n = new int[1];

string \*\*all\_ans = new string\*[10];

string \*\*PTran = new string\*[5];

private: System::Void AllTests\_Load(System::Object^ sender, System::EventArgs^ e) {

for (int i = 0; i < 10; i++) {

all\_ans[i] = new string[5];

}

n[0] = 0;

now\_test[0] = 0;

now\_test[1] = 0;

ans\_kol[0] = 0;

AllTest::all\_load(Test\_Types, PTran, vvod, Rbutton, panel1, panel2, panel3, panel7, dataGridView1, now\_qst, button12, ans\_kol, now\_test, all\_ans, n[0]);

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

System::Windows::Forms::DialogResult result = MessageBox::Show("Вы уверены что хотите завершить тест?\r\n(Вы увидите ответы только на полностью пройденные типы тестов!)", "Завершить тест", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (result == System::Windows::Forms::DialogResult::Yes) {

now\_test[0] = 10;

button1->Visible = false;

AllTest::test\_check(Test\_Types, now\_test, PTran, vvod, Rbutton, panel1, panel2, panel3, panel7, dataGridView1, all\_ans, n[0]);

}

}

private: System::Void trackBar1\_Scroll(System::Object^ sender, System::EventArgs^ e) {

AllTest::trackBar1\_changed(label1, PTran, ans\_kol, trackBar1, now\_qst, button2, button3, button4, button5, button6, button7, button8, button9, button10, button11, button12);

}

private: System::Void button11\_Click(System::Object^ sender, System::EventArgs^ e) {

AllTest::button11\_Press(PTran, ans\_kol, trackBar1, now\_qst, button2, button3, button4, button11, button12);

Achivments::Achivments\_get("3", 3, ConvertToString(Login\_Text));

}

private: System::Void button12\_Click(System::Object^ sender, System::EventArgs^ e) {

AllTest::button12\_Press(Test\_Types, now\_test, PTran, vvod, Rbutton, panel1, panel2, panel3, panel7, dataGridView1, ans\_kol, now\_qst, button12, button1, all\_ans, n);

}

private: System::Void panel1\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

AllTest::panel1\_Visible(label1, PTran, button2, button3, button4, button5, button6, button7, button8, button9, button10);

}

void BlockButton(String^Text, Button^B1, Button^B2, Button^B3) {

if (now\_qst[trackBar1->Value] == false)

AllTest\_Module::PhraseTranslate\_Module::FullCheck(Text, B1, B2, B3);

}

void BlockButton(Button^B) {

if (now\_qst[trackBar1->Value] == false)

AllTest\_Module::PhraseTranslate\_Module::RemoveEl(B);

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button5->Text, button2, button3, button4);

}

private: System::Void button6\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button6->Text, button2, button3, button4);

}

private: System::Void button7\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button7->Text, button2, button3, button4);

Achivments::Achivments\_get("3", 1, ConvertToString(Login\_Text));

}

private: System::Void button8\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button8->Text, button2, button3, button4);

}

private: System::Void button9\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button9->Text, button2, button3, button4);

}

private: System::Void button10\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button10->Text, button2, button3, button4);

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button2);

Achivments::Achivments\_get("2", 3, ConvertToString(Login\_Text));

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button3);

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

BlockButton(button4);

}

string \*\*vvod = new string\*[3];

private: System::Void panel2\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

AllTest::panel2\_Visible(label2, label3, textBox1, vvod, trackBar2->Value);

}

private: System::Void trackBar2\_Scroll(System::Object^ sender, System::EventArgs^ e) {

AllTest::trackBar2\_changed(label2, label3, textBox1, vvod, ans\_kol, trackBar2->Value, now\_qst, button12, button13);

}

private: System::Void button13\_Click(System::Object^ sender, System::EventArgs^ e) {

AllTest::button13\_Press(label3, textBox1, vvod, ans\_kol, trackBar2->Value, now\_qst, button12, button13);

}

string \*\*Rbutton = new string\*[7];

private: System::Void panel3\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

AllTest::panel3\_Visible(label4, label5, Rbutton, trackBar3->Value, radioButton1, radioButton2, radioButton3, radioButton4);

}

private: System::Void button14\_Click(System::Object^ sender, System::EventArgs^ e) {

AllTest::button14\_Press(label5, Rbutton, ans\_kol, trackBar3->Value, now\_qst, radioButton1, radioButton2, radioButton3, radioButton4, button12, button14, groupBox1);

}

private: System::Void trackBar3\_Scroll(System::Object^ sender, System::EventArgs^ e) {

AllTest::trackBar3\_change(label4, label5, Rbutton, ans\_kol, trackBar3->Value, now\_qst, radioButton1, radioButton2, radioButton3, radioButton4, button12, button14, groupBox1);

}

private: System::Void button22\_Click(System::Object^ sender, System::EventArgs^ e) {

Owner->Show();

this->Hide();

Achivments::Achivments\_get("3", 2, ConvertToString(Login\_Text));

}

private: System::Void AllTests\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

};

}

**Cabinet.h**

#pragma once

#include "VAR.h"

#include "TestsHeaders.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

using namespace Data::OleDb;

using namespace Obj\_Space;

/// <summary>

/// Сводка для Cabinet

/// </summary>

public ref class Cabinet : public System::Windows::Forms::Form

{

public:

Cabinet(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Cabinet()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Label^ label1;

protected:

private: System::Windows::Forms::PictureBox^ pictureBox1;

private: System::Windows::Forms::CheckBox^ checkBox1;

private: System::Windows::Forms::CheckBox^ checkBox2;

private: System::Windows::Forms::PictureBox^ pictureBox2;

private: System::Windows::Forms::PictureBox^ pictureBox3;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Timer^ CabinetTimer;

private: System::Windows::Forms::LinkLabel^ linkLabel1;

private: System::Windows::Forms::PictureBox^ pictureBox4;

private: System::Windows::Forms::Panel^ panel1;

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->label1 = (gcnew System::Windows::Forms::Label());

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->checkBox1 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox2 = (gcnew System::Windows::Forms::CheckBox());

this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox3 = (gcnew System::Windows::Forms::PictureBox());

this->label2 = (gcnew System::Windows::Forms::Label());

this->CabinetTimer = (gcnew System::Windows::Forms::Timer(this->components));

this->linkLabel1 = (gcnew System::Windows::Forms::LinkLabel());

this->pictureBox4 = (gcnew System::Windows::Forms::PictureBox());

this->panel1 = (gcnew System::Windows::Forms::Panel());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox2))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox3))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox4))->BeginInit();

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 19.8F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(171, 80);

this->label1->Margin = System::Windows::Forms::Padding(2, 0, 2, 0);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(95, 32);

this->label1->TabIndex = 0;

this->label1->Text = L"label1";

//

// pictureBox1

//

this->pictureBox1->Location = System::Drawing::Point(34, 80);

this->pictureBox1->Margin = System::Windows::Forms::Padding(2);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(102, 110);

this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox1->TabIndex = 2;

this->pictureBox1->TabStop = false;

//

// checkBox1

//

this->checkBox1->AutoSize = true;

this->checkBox1->FlatStyle = System::Windows::Forms::FlatStyle::Popup;

this->checkBox1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 13.8F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->checkBox1->Location = System::Drawing::Point(70, 22);

this->checkBox1->Margin = System::Windows::Forms::Padding(2);

this->checkBox1->Name = L"checkBox1";

this->checkBox1->Size = System::Drawing::Size(245, 26);

this->checkBox1->TabIndex = 3;

this->checkBox1->Text = L"Персональные данные";

this->checkBox1->UseVisualStyleBackColor = true;

//

// checkBox2

//

this->checkBox2->AutoSize = true;

this->checkBox2->BackColor = System::Drawing::SystemColors::Control;

this->checkBox2->FlatStyle = System::Windows::Forms::FlatStyle::Popup;

this->checkBox2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 13.8F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->checkBox2->Location = System::Drawing::Point(418, 25);

this->checkBox2->Margin = System::Windows::Forms::Padding(2);

this->checkBox2->Name = L"checkBox2";

this->checkBox2->Size = System::Drawing::Size(255, 26);

this->checkBox2->TabIndex = 4;

this->checkBox2->Text = L"Расписание экзаменов";

this->checkBox2->UseVisualStyleBackColor = false;

//

// pictureBox2

//

this->pictureBox2->Location = System::Drawing::Point(56, 19);

this->pictureBox2->Margin = System::Windows::Forms::Padding(2);

this->pictureBox2->Name = L"pictureBox2";

this->pictureBox2->Size = System::Drawing::Size(30, 32);

this->pictureBox2->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox2->TabIndex = 5;

this->pictureBox2->TabStop = false;

//

// pictureBox3

//

this->pictureBox3->Location = System::Drawing::Point(403, 22);

this->pictureBox3->Margin = System::Windows::Forms::Padding(2);

this->pictureBox3->Name = L"pictureBox3";

this->pictureBox3->Size = System::Drawing::Size(26, 28);

this->pictureBox3->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox3->TabIndex = 6;

this->pictureBox3->TabStop = false;

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 19.8F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->Location = System::Drawing::Point(29, 261);

this->label2->Margin = System::Windows::Forms::Padding(2, 0, 2, 0);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(96, 33);

this->label2->TabIndex = 7;

this->label2->Text = L"label2";

this->label2->Visible = false;

//

// CabinetTimer

//

this->CabinetTimer->Enabled = true;

this->CabinetTimer->Interval = 10;

this->CabinetTimer->Tick += gcnew System::EventHandler(this, &Cabinet::CabinetTimer\_Tick);

//

// linkLabel1

//

this->linkLabel1->AutoSize = true;

this->linkLabel1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 16.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->linkLabel1->LinkColor = System::Drawing::Color::Navy;

this->linkLabel1->Location = System::Drawing::Point(280, 526);

this->linkLabel1->Margin = System::Windows::Forms::Padding(2, 0, 2, 0);

this->linkLabel1->Name = L"linkLabel1";

this->linkLabel1->Size = System::Drawing::Size(282, 25);

this->linkLabel1->TabIndex = 9;

this->linkLabel1->TabStop = true;

this->linkLabel1->Text = L"Выйти из учетной записи";

this->linkLabel1->LinkClicked += gcnew System::Windows::Forms::LinkLabelLinkClickedEventHandler(this, &Cabinet::linkLabel1\_LinkClicked);

//

// pictureBox4

//

this->pictureBox4->Location = System::Drawing::Point(700, 13);

this->pictureBox4->Margin = System::Windows::Forms::Padding(2);

this->pictureBox4->Name = L"pictureBox4";

this->pictureBox4->Size = System::Drawing::Size(45, 49);

this->pictureBox4->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox4->TabIndex = 10;

this->pictureBox4->TabStop = false;

this->pictureBox4->Click += gcnew System::EventHandler(this, &Cabinet::pictureBox4\_Click);

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->pictureBox4);

this->panel1->Controls->Add(this->linkLabel1);

this->panel1->Controls->Add(this->label2);

this->panel1->Controls->Add(this->pictureBox3);

this->panel1->Controls->Add(this->pictureBox2);

this->panel1->Controls->Add(this->checkBox2);

this->panel1->Controls->Add(this->checkBox1);

this->panel1->Controls->Add(this->pictureBox1);

this->panel1->Controls->Add(this->label1);

this->panel1->Location = System::Drawing::Point(0, 1);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(783, 561);

this->panel1->TabIndex = 11;

//

// Cabinet

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(784, 561);

this->Controls->Add(this->panel1);

this->Margin = System::Windows::Forms::Padding(2);

this->Name = L"Cabinet";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Учётная запись";

this->Load += gcnew System::EventHandler(this, &Cabinet::Cabinet\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &Cabinet::Cabinet\_VisibleChanged);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox2))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox3))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox4))->EndInit();

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

this->ResumeLayout(false);

}

#pragma endregion

void Select\_Name(String ^ &ReturnText) {

auto p = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application::StartupPath + "\\resource\\Database1.mdb");

p->Open();

auto v = gcnew OleDbCommand("SELECT \* FROM [MAIN]", p);

v->CommandText = "SELECT [Имя] FROM [MAIN] WHERE [Login] = '" + ConvertToString(Login\_Text) + "'";

auto s = gcnew OleDbCommand("SELECT \* FROM [MAIN]", p);

s->CommandText = "SELECT [Фамилия] FROM [MAIN] WHERE [Login] = '" + ConvertToString(Login\_Text) + "'";

ReturnText = Convert::ToString(v->ExecuteScalar()) + " " + Convert::ToString(s->ExecuteScalar());

p->Close();

}

void Select\_DataPlus(String ^ &ReturnText) {

auto x = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application::StartupPath + "\\resource\\Database1.mdb");

x->Open();

auto y = gcnew OleDbCommand("SELECT \* FROM [MAIN]", x);

y->CommandText = "SELECT [Login] FROM [MAIN] WHERE [Login] = '" + ConvertToString(Login\_Text) + "'";

auto z = gcnew OleDbCommand("SELECT \* FROM [MAIN]", x);

z->CommandText = "SELECT [Группа] FROM [MAIN] WHERE [Login] = '" + ConvertToString(Login\_Text) + "'";

ReturnText = Convert::ToString(y->ExecuteScalar()) + " " + Convert::ToString(z->ExecuteScalar());

x->Close();

}

void Checkertron(){

if (checkBox1->Checked == 1 && checkBox2->Checked == 1) {

checkBox1->Checked = 0;

checkBox2->Checked = 0;

}

}

void CheckertronFunction(){

if (checkBox1->Checked == 1){

checkBox1->BackColor = System::Drawing::SystemColors::ScrollBar;

checkBox2->BackColor = System::Drawing::SystemColors::Control;

label2->Visible = 0;

}

if (checkBox2->Checked == 1){

checkBox2->BackColor = System::Drawing::SystemColors::ScrollBar;

checkBox1->BackColor = System::Drawing::SystemColors::Control;

int SelectDay = CalendarGlobal[0];

int SelectMonth = CalendarGlobal[1];

label2->Visible = 0;

label2->Text = "Вы сдаете экзамен " + Convert::ToString(SelectDay) + " . " + Convert::ToString(SelectMonth);

}

}

private: System::Void Cabinet\_Load(System::Object^ sender, System::EventArgs^ e){

pictureBox1->Image = Image::FromFile(Application::StartupPath + "\\UI\\CabinetPic.png");

pictureBox2->Image = Image::FromFile(Application::StartupPath + "\\UI\\Info.png");

pictureBox3->Image = Image::FromFile(Application::StartupPath + "\\UI\\When.png");

pictureBox4->Image = Image::FromFile(Application::StartupPath + "\\UI\\Arrow.png");

String ^ Name;

Select\_Name(Name);

label1->Text = Name;

String ^ DataPlus;

Select\_DataPlus(DataPlus);

label2->Text = DataPlus;

}

private: System::Void CabinetTimer\_Tick(System::Object^ sender, System::EventArgs^ e){

Checkertron();

CheckertronFunction();

}

private: System::Void linkLabel1\_LinkClicked(System::Object^ sender, System::Windows::Forms::LinkLabelLinkClickedEventArgs^ e) {

std::exit(0);

}

private: System::Void pictureBox4\_Click(System::Object^ sender, System::EventArgs^ e){

Owner->Show();

this->Hide();

}

private: System::Void Cabinet\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

};

}

**EntryForm.h**

#pragma once

#include "Start.h"

#include "TitulForm.h"

#include "VAR.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для EntryForm

/// </summary>

public ref class EntryForm : public System::Windows::Forms::Form

{

public:

EntryForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~EntryForm()

{

if (components)

{

delete components;

}

}

protected:

private: System::Windows::Forms::Timer^ EnterTimer;

private: System::Windows::Forms::Timer^ timer1;

private: System::Windows::Forms::Panel^ panel1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::PictureBox^ pictureBox1;

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(EntryForm::typeid));

this->EnterTimer = (gcnew System::Windows::Forms::Timer(this->components));

this->timer1 = (gcnew System::Windows::Forms::Timer(this->components));

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->panel1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

this->SuspendLayout();

this->EnterTimer->Tick += gcnew System::EventHandler(this, &EntryForm::EnterTimer\_Tick\_1);

this->timer1->Enabled = true;

this->timer1->Interval = 10;

this->timer1->Tick += gcnew System::EventHandler(this, &EntryForm::timer1\_Tick);

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->label2);

this->panel1->Controls->Add(this->label3);

this->panel1->Controls->Add(this->label1);

this->panel1->Controls->Add(this->pictureBox1);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(600, 280);

this->panel1->TabIndex = 5;

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->Location = System::Drawing::Point(16, 84);

this->label2->Margin = System::Windows::Forms::Padding(2, 0, 2, 0);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(566, 105);

this->label2->TabIndex = 12;

this->label2->Text = resources->GetString(L"label2.Text");

this->label2->Visible = false;

this->label2->Click += gcnew System::EventHandler(this, &EntryForm::label2\_Click);

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label3->ForeColor = System::Drawing::Color::Black;

this->label3->Location = System::Drawing::Point(29, 210);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(530, 24);

this->label3->TabIndex = 11;

this->label3->Text = L"Производится первичная настройка программы.";

this->label3->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;

this->label3->Visible = false;

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 36, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->ForeColor = System::Drawing::Color::White;

this->label1->Location = System::Drawing::Point(-8, 225);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(589, 116);

this->label1->TabIndex = 10;

this->label1->Text = L"Электронное пособие \r\nАнглийский язык\r\n";

this->label1->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;

this->pictureBox1->Location = System::Drawing::Point(270, 200);

this->pictureBox1->Margin = System::Windows::Forms::Padding(2);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(38, 41);

this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox1->TabIndex = 13;

this->pictureBox1->TabStop = false;

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(584, 261);

this->Controls->Add(this->panel1);

this->Name = L"EntryForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"EntryForm";

this->Load += gcnew System::EventHandler(this, &EntryForm::EntryForm\_Load);

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

this->ResumeLayout(false);

}

#pragma endregion

int Sec;

int LB1x = 7;

int LB1y = 224;

int WhiteIndex = 240;

int Index = 240;

int DotIndex = 0;

int BlueIndex = 0;

int BlueIndexBlack = 255;

void Mover2000(Label^LB, int x, int y) {

LB->Location = System::Drawing::Point(x, y);}

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e) {

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::None;}

private: System::Void timer1\_Tick(System::Object^ sender, System::EventArgs^ e){

DotIndex++;

if (DotIndex == 30)

DotIndex = 0;

Sec++;

if (Sec > 100) {

if (Index > 0) {

Index = Index - 2;

label3->ForeColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(Index)), static\_cast<System::Int32>(static\_cast<System::Byte>(Index)),

static\_cast<System::Int32>(static\_cast<System::Byte>(Index)));

}

}

if (DotIndex == 0 && Sec > 100 && Sec < 200)

label3->Text = "Производится первичная настройка программы.";

if (DotIndex == 10 && Sec > 100 && Sec < 200)

label3->Text = "Производится первичная настройка программы..";

if (DotIndex == 20 && Sec > 100 && Sec < 200)

label3->Text = "Производится первичная настройка программы...";

if (Sec == 200)

{

Mover2000(label3, 120, 210);

label3->Text = "База данных успешно создана";

}

if (Sec == 250)

{

Mover2000(label3, 70, 210);

label3->Text = "Структура базы данных успешно создана";

}

if (Sec == 300)

{

Mover2000(label3, 170, 210);

label3->Text = "Спасибо за ожидание";

}

if (Sec == 850)

{

pictureBox1->Image = Image::FromFile(Application::StartupPath + "\\UI\\Done.png");

}

if (Sec > 310 && Sec < 560)

{

BlueIndex = BlueIndex - 1;

panel1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(BlueIndex)), static\_cast<System::Int32>(static\_cast<System::Byte>(BlueIndex)),

static\_cast<System::Int32>(static\_cast<System::Byte>(BlueIndex)));

}

if (Sec > 570 && Sec < 820)

{

label1->Visible = 0;

label3->Visible = 0;

label2->Visible = 1;

BlueIndexBlack = BlueIndexBlack + 1;

panel1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(BlueIndexBlack)), static\_cast<System::Int32>(static\_cast<System::Byte>(BlueIndexBlack)),

static\_cast<System::Int32>(static\_cast<System::Byte>(BlueIndexBlack)));

}

if (Sec >= 900)

{

timer1->Enabled = false;

String^Path = Application::StartupPath + "\\resource\\DataBase1.mdb";

if (!IO::File::Exists(Path)) {

ADOX::Catalog ^K = gcnew ADOX::CatalogClass();

try

{

K->Create("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application::StartupPath + "\\resource\\Database1.mdb");

}

catch (System::Runtime::InteropServices::COMException ^ situation) {

bool ready = true;

MessageBox::Show(situation->Message);

}

finally

{

K = nullptr;

}

auto p = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application::StartupPath + "\\resource\\Database1.mdb");

p->Open();

auto c = gcnew OleDbCommand("CREATE TABLE [MAIN] ([ID] counter, [Login] String, [Password] String, [Имя] String, [Фамилия] String, [Группа] String, [Достижение 0] String, [Достижение 1] String, [Достижение 2] String, [Достижение 3] String, [Достижение 4] String)", p);

auto n = gcnew OleDbCommand("CREATE TABLE [SETTINGS] ([Size] String)", p);

try

{

n->ExecuteNonQuery();

c->ExecuteNonQuery();

}

catch (System::Runtime::InteropServices::COMException ^ situation) {

MessageBox::Show(situation->Message);

}

p->Close();

}

auto f = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application::StartupPath + "\\resource\\Database1.mdb");

f->Open();

auto v = gcnew OleDbCommand("SELECT \* FROM [SETTINGS]", f);

v->CommandText = "SELECT [Size] FROM [SETTINGS]";

if (Convert::ToString(v->ExecuteScalar()) == "") {

f->Close();

Start ^form = gcnew Start();

form->Owner = this;

form->Show();

this->Hide();

}

else {

f->Close();

Login ^form = gcnew Login();

form->Owner = this;

form->Show();

this->Hide();

}

}

if (Sec == 150)

{

label3->Visible = 1;

}

if (LB1y > 70)

{

LB1y = LB1y - 2;

}

if (WhiteIndex > 0)

{

WhiteIndex = WhiteIndex - 2;

}

Mover2000(label1, LB1x, LB1y);

label1->ForeColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(WhiteIndex)), static\_cast<System::Int32>(static\_cast<System::Byte>(WhiteIndex)),

static\_cast<System::Int32>(static\_cast<System::Byte>(WhiteIndex)));

}

private: System::Void EntryForm\_Load(System::Object^ sender, System::EventArgs^ e)

{

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::None;

}

private: System::Void EnterTimer\_Tick\_1(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void label2\_Click(System::Object^ sender, System::EventArgs^ e) {

}

};

}

**Login.h**

#pragma once

#include <cctype>

#include "TitulForm.h";

#include "VAR.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

using namespace std;

/// <summary>

/// Сводка для Login

/// </summary>

public ref class Login : public System::Windows::Forms::Form

{

public:

Login(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Login()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::TextBox^ textBox1;

protected:

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Panel^ panel1;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::LinkLabel^ linkLabel1;

private: System::Windows::Forms::Panel^ panel2;

private: System::Windows::Forms::Label^ label7;

private: System::Windows::Forms::TextBox^ textBox7;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::Label^ label6;

private: System::Windows::Forms::TextBox^ textBox5;

private: System::Windows::Forms::TextBox^ textBox6;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::TextBox^ textBox4;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::PictureBox^ pictureBox1;

private: System::Windows::Forms::PictureBox^ pictureBox2;

private: System::Windows::Forms::PictureBox^ pictureBox3;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->button2 = (gcnew System::Windows::Forms::Button());

this->linkLabel1 = (gcnew System::Windows::Forms::LinkLabel());

this->button1 = (gcnew System::Windows::Forms::Button());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->panel2 = (gcnew System::Windows::Forms::Panel());

this->pictureBox3 = (gcnew System::Windows::Forms::PictureBox());

this->label7 = (gcnew System::Windows::Forms::Label());

this->textBox7 = (gcnew System::Windows::Forms::TextBox());

this->label5 = (gcnew System::Windows::Forms::Label());

this->label6 = (gcnew System::Windows::Forms::Label());

this->textBox5 = (gcnew System::Windows::Forms::TextBox());

this->textBox6 = (gcnew System::Windows::Forms::TextBox());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->textBox4 = (gcnew System::Windows::Forms::TextBox());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->button3 = (gcnew System::Windows::Forms::Button());

this->panel1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox2))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

this->panel2->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox3))->BeginInit();

this->SuspendLayout();

//

// textBox1

//

this->textBox1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox1->Location = System::Drawing::Point(259, 193);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(245, 33);

this->textBox1->TabIndex = 0;

//

// textBox2

//

this->textBox2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox2->Location = System::Drawing::Point(259, 254);

this->textBox2->Name = L"textBox2";

this->textBox2->PasswordChar = '\*';

this->textBox2->Size = System::Drawing::Size(245, 33);

this->textBox2->TabIndex = 1;

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->BackColor = System::Drawing::SystemColors::Control;

this->panel1->Controls->Add(this->pictureBox2);

this->panel1->Controls->Add(this->pictureBox1);

this->panel1->Controls->Add(this->button2);

this->panel1->Controls->Add(this->linkLabel1);

this->panel1->Controls->Add(this->button1);

this->panel1->Controls->Add(this->label2);

this->panel1->Controls->Add(this->label1);

this->panel1->Controls->Add(this->textBox2);

this->panel1->Controls->Add(this->textBox1);

this->panel1->Location = System::Drawing::Point(12, 12);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(760, 537);

this->panel1->TabIndex = 3;

this->panel1->Visible = false;

//

// pictureBox2

//

this->pictureBox2->Location = System::Drawing::Point(384, 115);

this->pictureBox2->Name = L"pictureBox2";

this->pictureBox2->Size = System::Drawing::Size(60, 60);

this->pictureBox2->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox2->TabIndex = 10;

this->pictureBox2->TabStop = false;

//

// pictureBox1

//

this->pictureBox1->Location = System::Drawing::Point(330, 43);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(100, 100);

this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox1->TabIndex = 9;

this->pictureBox1->TabStop = false;

//

// button2

//

this->button2->BackColor = System::Drawing::Color::RoyalBlue;

this->button2->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button2->Location = System::Drawing::Point(643, 22);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(98, 43);

this->button2->TabIndex = 8;

this->button2->Text = L"Выход";

this->button2->UseVisualStyleBackColor = false;

this->button2->Click += gcnew System::EventHandler(this, &Login::button2\_Click);

//

// linkLabel1

//

this->linkLabel1->AutoSize = true;

this->linkLabel1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->linkLabel1->Location = System::Drawing::Point(273, 302);

this->linkLabel1->Name = L"linkLabel1";

this->linkLabel1->Size = System::Drawing::Size(227, 24);

this->linkLabel1->TabIndex = 7;

this->linkLabel1->TabStop = true;

this->linkLabel1->Text = L"Зарегистрироваться";

this->linkLabel1->LinkClicked += gcnew System::Windows::Forms::LinkLabelLinkClickedEventHandler(this, &Login::linkLabel1\_LinkClicked);

//

// button1

//

this->button1->BackColor = System::Drawing::Color::RoyalBlue;

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button1->Location = System::Drawing::Point(226, 396);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(300, 80);

this->button1->TabIndex = 5;

this->button1->Text = L"Авторизоваться";

this->button1->UseVisualStyleBackColor = false;

this->button1->Click += gcnew System::EventHandler(this, &Login::button1\_Click);

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->Location = System::Drawing::Point(336, 227);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(88, 24);

this->label2->TabIndex = 4;

this->label2->Text = L"Пароль";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(345, 167);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(73, 24);

this->label1->TabIndex = 3;

this->label1->Text = L"Логин";

//

// panel2

//

this->panel2->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel2->Controls->Add(this->pictureBox3);

this->panel2->Controls->Add(this->label7);

this->panel2->Controls->Add(this->textBox7);

this->panel2->Controls->Add(this->label5);

this->panel2->Controls->Add(this->label6);

this->panel2->Controls->Add(this->textBox5);

this->panel2->Controls->Add(this->textBox6);

this->panel2->Controls->Add(this->label3);

this->panel2->Controls->Add(this->label4);

this->panel2->Controls->Add(this->textBox4);

this->panel2->Controls->Add(this->textBox3);

this->panel2->Controls->Add(this->button3);

this->panel2->Location = System::Drawing::Point(12, 12);

this->panel2->Name = L"panel2";

this->panel2->Size = System::Drawing::Size(760, 537);

this->panel2->TabIndex = 4;

this->panel2->Visible = false;

//

// pictureBox3

//

this->pictureBox3->Location = System::Drawing::Point(309, 33);

this->pictureBox3->Name = L"pictureBox3";

this->pictureBox3->Size = System::Drawing::Size(135, 116);

this->pictureBox3->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox3->TabIndex = 11;

this->pictureBox3->TabStop = false;

//

// label7

//

this->label7->AutoSize = true;

this->label7->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label7->Location = System::Drawing::Point(510, 262);

this->label7->Name = L"label7";

this->label7->Size = System::Drawing::Size(85, 24);

this->label7->TabIndex = 15;

this->label7->Text = L"Группа";

//

// textBox7

//

this->textBox7->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox7->Location = System::Drawing::Point(424, 289);

this->textBox7->Name = L"textBox7";

this->textBox7->Size = System::Drawing::Size(245, 31);

this->textBox7->TabIndex = 14;

//

// label5

//

this->label5->AutoSize = true;

this->label5->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label5->Location = System::Drawing::Point(500, 201);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(107, 24);

this->label5->TabIndex = 13;

this->label5->Text = L"Фамилия";

//

// label6

//

this->label6->AutoSize = true;

this->label6->Font = (gcnew System::Drawing::Font(L"Century Gothic", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label6->Location = System::Drawing::Point(527, 140);

this->label6->Name = L"label6";

this->label6->Size = System::Drawing::Size(53, 24);

this->label6->TabIndex = 12;

this->label6->Text = L"Имя";

//

// textBox5

//

this->textBox5->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox5->Location = System::Drawing::Point(424, 167);

this->textBox5->Name = L"textBox5";

this->textBox5->Size = System::Drawing::Size(245, 31);

this->textBox5->TabIndex = 11;

//

// textBox6

//

this->textBox6->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox6->Location = System::Drawing::Point(424, 228);

this->textBox6->Name = L"textBox6";

this->textBox6->Size = System::Drawing::Size(245, 31);

this->textBox6->TabIndex = 10;

//

// label3

//

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label3->Location = System::Drawing::Point(148, 239);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(115, 33);

this->label3->TabIndex = 9;

this->label3->Text = L"Пароль";

//

// label4

//

this->label4->AutoSize = true;

this->label4->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label4->Location = System::Drawing::Point(148, 157);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(95, 33);

this->label4->TabIndex = 8;

this->label4->Text = L"Логин";

//

// textBox4

//

this->textBox4->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox4->Location = System::Drawing::Point(83, 275);

this->textBox4->Name = L"textBox4";

this->textBox4->Size = System::Drawing::Size(245, 31);

this->textBox4->TabIndex = 2;

//

// textBox3

//

this->textBox3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox3->Location = System::Drawing::Point(83, 193);

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(245, 31);

this->textBox3->TabIndex = 1;

//

// button3

//

this->button3->BackColor = System::Drawing::Color::DodgerBlue;

this->button3->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button3->Location = System::Drawing::Point(226, 396);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(300, 80);

this->button3->TabIndex = 0;

this->button3->Text = L"Зарегистрироваться";

this->button3->UseVisualStyleBackColor = false;

this->button3->Click += gcnew System::EventHandler(this, &Login::button3\_Click);

//

// Login

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 17);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(784, 561);

this->ControlBox = false;

this->Controls->Add(this->panel1);

this->Controls->Add(this->panel2);

this->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"Login";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Учётная запись";

this->Load += gcnew System::EventHandler(this, &Login::Login\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &Login::Login\_VisibleChanged);

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox2))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

this->panel2->ResumeLayout(false);

this->panel2->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox3))->EndInit();

this->ResumeLayout(false);

}

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

TitulForm ^ form = gcnew TitulForm();

Login\_Text = ConvertTostring(textBox1->Text);

Login\_Module::LoginEnter(textBox1, textBox2, form, this);

}

private: System::Void linkLabel1\_LinkClicked(System::Object^ sender, System::Windows::Forms::LinkLabelLinkClickedEventArgs^ e) {

textBox1->Text = "";

textBox2->Text = "";

panel1->Visible = false;

panel2->Visible = true;

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

Login\_Module::LoginReg(textBox3, textBox4, textBox5, textBox6, textBox7, panel1, panel2);

}

private: System::Void Login\_Load(System::Object^ sender, System::EventArgs^ e) {

panel1->Visible = true;

panel2->Visible = false;

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

exit(0);

}

private: System::Void Login\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

};

}

**MainMenuForm.h**

#pragma once

#include "Teor.h"

#include "TestSelect.h"

#include "Teor2.h"

#include "ProgressForm.h"

#include "Settings.h"

#include "Cabinet.h"

#include "AchivmentsForm.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

/// <summary>

/// Сводка для MainMenuForm

/// </summary>

public ref class MainMenuForm : public System::Windows::Forms::Form

{

public:

MainMenuForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MainMenuForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button10;

protected:

private: System::Windows::Forms::Button^ button9;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Button^ button6;

private: System::Windows::Forms::Button^ button7;

private: System::Windows::Forms::Button^ button8;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Timer^ ButtonMoverTimer;

private: System::Windows::Forms::Timer^ TestsTimer;

private: System::Windows::Forms::Timer^ CabinetTimer;

private: System::Windows::Forms::Timer^ OptionTimer;

private: System::Windows::Forms::Panel^ panel1;

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->button10 = (gcnew System::Windows::Forms::Button());

this->button9 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->button6 = (gcnew System::Windows::Forms::Button());

this->button7 = (gcnew System::Windows::Forms::Button());

this->button8 = (gcnew System::Windows::Forms::Button());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->ButtonMoverTimer = (gcnew System::Windows::Forms::Timer(this->components));

this->TestsTimer = (gcnew System::Windows::Forms::Timer(this->components));

this->CabinetTimer = (gcnew System::Windows::Forms::Timer(this->components));

this->OptionTimer = (gcnew System::Windows::Forms::Timer(this->components));

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// button10

//

this->button10->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button10->Font = (gcnew System::Drawing::Font(L"Century Gothic", 21.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button10->Location = System::Drawing::Point(420, 310);

this->button10->Name = L"button10";

this->button10->Size = System::Drawing::Size(200, 200);

this->button10->TabIndex = 21;

this->button10->UseVisualStyleBackColor = true;

this->button10->Click += gcnew System::EventHandler(this, &MainMenuForm::button10\_Click);

//

// button9

//

this->button9->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button9->Font = (gcnew System::Drawing::Font(L"Century Gothic", 21.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button9->Location = System::Drawing::Point(160, 310);

this->button9->Name = L"button9";

this->button9->Size = System::Drawing::Size(200, 200);

this->button9->TabIndex = 20;

this->button9->UseVisualStyleBackColor = true;

this->button9->Click += gcnew System::EventHandler(this, &MainMenuForm::button9\_Click\_1);

//

// button5

//

this->button5->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button5->Font = (gcnew System::Drawing::Font(L"Century Gothic", 21.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button5->Location = System::Drawing::Point(420, 62);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(200, 200);

this->button5->TabIndex = 16;

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &MainMenuForm::button5\_Click\_1);

//

// button6

//

this->button6->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button6->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button6->Location = System::Drawing::Point(420, 62);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(200, 200);

this->button6->TabIndex = 17;

this->button6->Text = L"Календарь";

this->button6->UseVisualStyleBackColor = true;

this->button6->Click += gcnew System::EventHandler(this, &MainMenuForm::button6\_Click);

//

// button7

//

this->button7->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button7->Font = (gcnew System::Drawing::Font(L"Century Gothic", 21.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button7->Location = System::Drawing::Point(420, 64);

this->button7->Name = L"button7";

this->button7->Size = System::Drawing::Size(200, 200);

this->button7->TabIndex = 18;

this->button7->Text = L"Учетная запись";

this->button7->UseVisualStyleBackColor = true;

this->button7->Click += gcnew System::EventHandler(this, &MainMenuForm::button7\_Click);

//

// button8

//

this->button8->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button8->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button8->Location = System::Drawing::Point(420, 60);

this->button8->Name = L"button8";

this->button8->Size = System::Drawing::Size(200, 200);

this->button8->TabIndex = 19;

this->button8->Text = L"Достижения";

this->button8->UseVisualStyleBackColor = true;

this->button8->Click += gcnew System::EventHandler(this, &MainMenuForm::button8\_Click);

//

// button1

//

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 21.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button1->Location = System::Drawing::Point(160, 60);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(200, 200);

this->button1->TabIndex = 11;

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MainMenuForm::button1\_Click\_1);

//

// button2

//

this->button2->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 21.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button2->Location = System::Drawing::Point(160, 60);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(200, 200);

this->button2->TabIndex = 12;

this->button2->Text = L"Теория";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &MainMenuForm::button2\_Click\_2);

//

// button3

//

this->button3->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button3->Location = System::Drawing::Point(160, 60);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(200, 200);

this->button3->TabIndex = 14;

this->button3->Text = L"Тестирование";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &MainMenuForm::button3\_Click);

//

// button4

//

this->button4->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button4->Font = (gcnew System::Drawing::Font(L"Century Gothic", 21.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button4->Location = System::Drawing::Point(160, 62);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(200, 200);

this->button4->TabIndex = 13;

this->button4->Text = L"Резюме";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &MainMenuForm::button4\_Click);

//

// ButtonMoverTimer

//

this->ButtonMoverTimer->Enabled = true;

this->ButtonMoverTimer->Interval = 1;

this->ButtonMoverTimer->Tick += gcnew System::EventHandler(this, &MainMenuForm::ButtonMoverTimer\_Tick);

//

// TestsTimer

//

this->TestsTimer->Enabled = true;

this->TestsTimer->Interval = 1;

this->TestsTimer->Tick += gcnew System::EventHandler(this, &MainMenuForm::TestsTimer\_Tick\_1);

//

// CabinetTimer

//

this->CabinetTimer->Enabled = true;

this->CabinetTimer->Interval = 1;

this->CabinetTimer->Tick += gcnew System::EventHandler(this, &MainMenuForm::CabinetTimer\_Tick\_1);

//

// OptionTimer

//

this->OptionTimer->Enabled = true;

this->OptionTimer->Interval = 1;

this->OptionTimer->Tick += gcnew System::EventHandler(this, &MainMenuForm::OptionTimer\_Tick\_1);

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->button10);

this->panel1->Controls->Add(this->button9);

this->panel1->Controls->Add(this->button5);

this->panel1->Controls->Add(this->button6);

this->panel1->Controls->Add(this->button7);

this->panel1->Controls->Add(this->button8);

this->panel1->Controls->Add(this->button1);

this->panel1->Controls->Add(this->button2);

this->panel1->Controls->Add(this->button3);

this->panel1->Controls->Add(this->button4);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(784, 560);

this->panel1->TabIndex = 22;

//

// MainMenuForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(784, 561);

this->Controls->Add(this->panel1);

this->Name = L"MainMenuForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Главное меню";

this->Load += gcnew System::EventHandler(this, &MainMenuForm::MainMenuForm\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &MainMenuForm::MainMenuForm\_VisibleChanged);

this->panel1->ResumeLayout(false);

this->ResumeLayout(false);

}

#pragma endregion

int sec;

int TestsSec;

int B2y = 60;

int B3x = 160;

int B4x = 160;

int B4y = 60;

int B7x = 420;

int B6y = 60;

int B8x = 420;

int B8y = 60;

int OptionSec;

int Button1EnableIndex = 0;

int MoveIndex = 0;

int Button2EnableIndex = 0;

int Button3EnableIndex = 0;

int MoveIndex2 = 0;

int MoveIndex3 = 0;

int CabinetSec;

int B9x = 160;

int B9y = 310;

int W = 200;

int H = 200;

void Mover2000(Button^BT, int x, int y){

BT->Location = System::Drawing::Point(x, y);

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

ButtonMoverTimer->Enabled = 1;

TestsTimer->Enabled = 1;

TestsSec = 0;

if (Button1EnableIndex == 1)

MoveIndex = 1;

}

private: System::Void timer1\_Tick(System::Object^ sender, System::EventArgs^ e){

sec++;

}

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e){

ButtonMoverTimer->Enabled = 0;

TestsTimer->Enabled = 0;

CabinetTimer->Enabled = 0;

OptionTimer->Enabled = 0;

}

private: System::Void TestsTimer\_Tick(System::Object^ sender, System::EventArgs^ e)

{

TestsSec++;

if (MoveIndex == 0){

if (B2y <= 310){

B2y = B2y + 10;

Mover2000(button2, 160, B2y);

}

if (B3x <= 420){

B3x = B3x + 10;

Mover2000(button3, B3x, 60);

}

if (B4x <= 420 && B4y <= 310){

B4x = B4x + 10;

B4y = B4y + 10;

Mover2000(button4, B4x, B4y);

button1->BackColor = System::Drawing::Color::LightGreen;

button2->BackColor = System::Drawing::Color::LightGreen;

button3->BackColor = System::Drawing::Color::LightGreen;

button4->BackColor = System::Drawing::Color::LightGreen;

button5->Visible = 0;

button6->Visible = 0;

button7->Visible = 0;

button8->Visible = 0;

button9->Visible = 0;

button10->Visible = 0;

}

if (TestsSec > 10){

button1->Text = "";

Button1EnableIndex = 1;

}

}

if (MoveIndex == 1){

if (B2y > 60){

B2y = B2y - 10;

Mover2000(button2, 160, B2y);

}

if (B3x > 160){

B3x = B3x - 10;

Mover2000(button3, B3x, 60);

}

if (B4x > 160 && B4y > 60){

B4x = B4x - 10;

B4y = B4y - 10;

Mover2000(button4, B4x, B4y);

button1->BackColor = System::Drawing::SystemColors::Control;

button2->BackColor = System::Drawing::SystemColors::Control;

button3->BackColor = System::Drawing::SystemColors::Control;

button4->BackColor = System::Drawing::SystemColors::Control;

button5->Visible = 1;

button6->Visible = 1;

button7->Visible = 1;

button8->Visible = 1;

button9->Visible = 1;

button10->Visible = 1;

}

if (TestsSec == 30 && MoveIndex == 1){

Button1EnableIndex = 0;

MoveIndex = 0;

TestsTimer->Enabled = 0;

button1->Text = "";

}

}

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e){

CabinetTimer->Enabled = 1;

CabinetSec = 0;

if (Button2EnableIndex == 1)

MoveIndex2 = 1;

}

private: System::Void CabinetTimer\_Tick(System::Object^ sender, System::EventArgs^ e){

CabinetSec++;

if (MoveIndex2 == 0){

if (B6y <= 310){

B6y = B6y + 10;

Mover2000(button6, 420, B6y);

}

if (B7x > 160){

B7x = B7x - 10;

Mover2000(button7, B7x, 60);

}

if (B8x >= 160 && B8y <= 310){

B8x = B8x - 10;

B8y = B8y + 10;

Mover2000(button8, B8x, B8y);

button5->BackColor = System::Drawing::Color::SkyBlue;

button6->BackColor = System::Drawing::Color::SkyBlue;

button7->BackColor = System::Drawing::Color::SkyBlue;

button8->BackColor = System::Drawing::Color::SkyBlue;

button1->Visible = 0;

button2->Visible = 0;

button3->Visible = 0;

button4->Visible = 0;

button9->Visible = 0;

button10->Visible = 0;

}

if (CabinetSec > 10){

button5->Text = "";

Button2EnableIndex = 1;

}

}

if (MoveIndex2 == 1){

if (B7x < 420){

B7x = B7x + 10;

Mover2000(button7, B7x, 60);

}

if (B6y > 60){

B6y = B6y - 10;

Mover2000(button6, 420, B6y);

}

if (B8x < 420 && B8y > 60){

B8x = B8x + 10;

B8y = B8y - 10;

Mover2000(button8, B8x, B8y);

button1->Visible = 1;

button2->Visible = 1;

button3->Visible = 1;

button4->Visible = 1;

button9->Visible = 1;

button10->Visible = 1;

button5->BackColor = System::Drawing::SystemColors::Control;

button6->BackColor = System::Drawing::SystemColors::Control;

button7->BackColor = System::Drawing::SystemColors::Control;

button8->BackColor = System::Drawing::SystemColors::Control;

}

if (CabinetSec == 30 && MoveIndex2 == 1){

Button2EnableIndex = 0;

MoveIndex2 = 0;

CabinetTimer->Enabled = 0;

button5->Text = "";

}

}

}

private: System::Void button9\_Click(System::Object^ sender, System::EventArgs^ e){

OptionTimer->Enabled = 1;

OptionSec = 0;

if (Button3EnableIndex == 1)

MoveIndex3 = 1;

}

private: System::Void OptionTimer\_Tick(System::Object^ sender, System::EventArgs^ e){

OptionSec++;

if (MoveIndex3 == 0){

button9->BackColor = System::Drawing::Color::DarkOrange;

button1->Visible = 0;

button2->Visible = 0;

button3->Visible = 0;

button4->Visible = 0;

button5->Visible = 0;

button6->Visible = 0;

button7->Visible = 0;

button8->Visible = 0;

button10->Visible = 0;

if (H <= 460 && W < 460 && B9y > 60){

H = H + 10;

W = W + 10;

B9y = B9y - 10;

button9->Size = System::Drawing::Size(W, H);

Mover2000(button9, 160, B9y);

}

if (OptionSec > 10){

button9->Text = " ";

Button3EnableIndex = 1;

}

}

if (MoveIndex3 == 1){

if (H > 200 && W > 200 && B9y < 310){

button9->BackColor = System::Drawing::SystemColors::Control;

button1->Visible = 1;

button2->Visible = 1;

button3->Visible = 1;

button4->Visible = 1;

button5->Visible = 1;

button6->Visible = 1;

button7->Visible = 1;

button8->Visible = 1;

button10->Visible = 1;

H = H - 10;

W = W - 10;

B9y = B9y + 10;

button9->Size = System::Drawing::Size(W, H);

Mover2000(button9, 160, B9y);

}

if (OptionSec == 30 && MoveIndex3 == 1){

Button3EnableIndex = 0;

MoveIndex3 = 0;

OptionTimer->Enabled = 0;

button9->Text = " ";

}

}

}

private: System::Void ButtonMoverTimer\_Tick(System::Object^ sender, System::EventArgs^ e) {}

private: System::Void TestsTimer\_Tick\_1(System::Object^ sender, System::EventArgs^ e) {

TestsSec++;

if (MoveIndex == 0){

if (B2y <= 310){

B2y = B2y + 10;

Mover2000(button2, 160, B2y);

}

if (B3x <= 420){

B3x = B3x + 10;

Mover2000(button3, B3x, 60);

}

if (B4x <= 420 && B4y <= 310){

B4x = B4x + 10;

B4y = B4y + 10;

Mover2000(button4, B4x, B4y);

button1->BackColor = System::Drawing::Color::LimeGreen;

button2->BackColor = System::Drawing::Color::LimeGreen;

button3->BackColor = System::Drawing::Color::LimeGreen;

button4->BackColor = System::Drawing::Color::LimeGreen;

button5->Visible = 0;

button6->Visible = 0;

button7->Visible = 0;

button8->Visible = 0;

button9->Visible = 0;

button10->Visible = 0;

}

if (TestsSec > 10){

button1->Text = "";

Button1EnableIndex = 1;

}

}

if (MoveIndex == 1){

if (B2y > 60){

B2y = B2y - 10;

Mover2000(button2, 160, B2y);

}

if (B3x > 160){

B3x = B3x - 10;

Mover2000(button3, B3x, 60);

}

if (B4x > 160 && B4y > 60){

B4x = B4x - 10;

B4y = B4y - 10;

Mover2000(button4, B4x, B4y);

button1->BackColor = System::Drawing::SystemColors::Control;

button2->BackColor = System::Drawing::SystemColors::Control;

button3->BackColor = System::Drawing::SystemColors::Control;

button4->BackColor = System::Drawing::SystemColors::Control;

button5->Visible = 1;

button6->Visible = 1;

button7->Visible = 1;

button8->Visible = 1;

button9->Visible = 1;

button10->Visible = 1;

}

if (TestsSec == 30 && MoveIndex == 1){

Button1EnableIndex = 0;

MoveIndex = 0;

TestsTimer->Enabled = 0;

button1->Text = "";

}

}

}

private: System::Void CabinetTimer\_Tick\_1(System::Object^ sender, System::EventArgs^ e)

{

CabinetSec++;

if (MoveIndex2 == 0){

if (B6y <= 310){

B6y = B6y + 10;

Mover2000(button6, 420, B6y);

}

if (B7x > 160){

B7x = B7x - 10;

Mover2000(button7, B7x, 60);

}

if (B8x >= 160 && B8y <= 310){

B8x = B8x - 10;

B8y = B8y + 10;

Mover2000(button8, B8x, B8y);

button5->BackColor = System::Drawing::Color::DodgerBlue;

button6->BackColor = System::Drawing::Color::DodgerBlue;

button7->BackColor = System::Drawing::Color::DodgerBlue;

button8->BackColor = System::Drawing::Color::DodgerBlue;

button1->Visible = 0;

button2->Visible = 0;

button3->Visible = 0;

button4->Visible = 0;

button9->Visible = 0;

button10->Visible = 0;

}

if (CabinetSec > 10){

button5->Text = "";

Button2EnableIndex = 1;

}

}

if (MoveIndex2 == 1){

if (B7x < 420){

B7x = B7x + 10;

Mover2000(button7, B7x, 60);

}

if (B6y > 60){

B6y = B6y - 10;

Mover2000(button6, 420, B6y);

}

if (B8x < 420 && B8y > 60){

B8x = B8x + 10;

B8y = B8y - 10;

Mover2000(button8, B8x, B8y);

button1->Visible = 1;

button2->Visible = 1;

button3->Visible = 1;

button4->Visible = 1;

button9->Visible = 1;

button10->Visible = 1;

button5->BackColor = System::Drawing::SystemColors::Control;

button6->BackColor = System::Drawing::SystemColors::Control;

button7->BackColor = System::Drawing::SystemColors::Control;

button8->BackColor = System::Drawing::SystemColors::Control;

}

if (CabinetSec == 30 && MoveIndex2 == 1){

Button2EnableIndex = 0;

MoveIndex2 = 0;

CabinetTimer->Enabled = 0;

button5->Text = "";

}

}

}

private: System::Void OptionTimer\_Tick\_1(System::Object^ sender, System::EventArgs^ e)

{

OptionSec++;

if (MoveIndex3 == 0){

button9->BackColor = System::Drawing::Color::Orange;

button1->Visible = 0;

button2->Visible = 0;

button3->Visible = 0;

button4->Visible = 0;

button5->Visible = 0;

button6->Visible = 0;

button7->Visible = 0;

button8->Visible = 0;

button10->Visible = 0;

if (H <= 460 && W < 460 && B9y > 60){

H = H + 10;

W = W + 10;

B9y = B9y - 10;

button9->Size = System::Drawing::Size(W, H);

Mover2000(button9, 160, B9y);

}

if (OptionSec > 10){

button9->Text = " ";

Button3EnableIndex = 1;

}

}

if (MoveIndex3 == 1){

if (H > 200 && W > 200 && B9y < 310){

button9->BackColor = System::Drawing::SystemColors::Control;

button1->Visible = 1;

button2->Visible = 1;

button3->Visible = 1;

button4->Visible = 1;

button5->Visible = 1;

button6->Visible = 1;

button7->Visible = 1;

button8->Visible = 1;

button10->Visible = 1;

H = H - 10;

W = W - 10;

B9y = B9y + 10;

button9->Size = System::Drawing::Size(W, H);

Mover2000(button9, 160, B9y);

}

if (OptionSec == 30 && MoveIndex3 == 1){

Button3EnableIndex = 0;

MoveIndex3 = 0;

OptionTimer->Enabled = 0;

button9->Text = " ";

}

}

}

private: System::Void MainMenuForm\_Load(System::Object^ sender, System::EventArgs^ e)

{

button1->Image = Image::FromFile(Application::StartupPath + "\\UI\\Obuchenie.png");

button5->Image = Image::FromFile(Application::StartupPath + "\\UI\\Cabinet.png");

button9->Image = Image::FromFile(Application::StartupPath + "\\UI\\Settings.png");

button10->Image = Image::FromFile(Application::StartupPath + "\\UI\\Exit.png");

OptionTimer->Enabled = 0;

CabinetTimer->Enabled = 0;

TestsTimer->Enabled = 0;

}

private: System::Void button1\_Click\_1(System::Object^ sender, System::EventArgs^ e) {

ButtonMoverTimer->Enabled = 1;

TestsTimer->Enabled = 1;

TestsSec = 0;

if (Button1EnableIndex == 1)

{

MoveIndex = 1;

}

}

private: System::Void button9\_Click\_1(System::Object^ sender, System::EventArgs^ e){

Settings ^form = gcnew Settings();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button5\_Click\_1(System::Object^ sender, System::EventArgs^ e){

CabinetTimer->Enabled = 1;

CabinetSec = 0;

if (Button2EnableIndex == 1)

MoveIndex2 = 1;

}

private: System::Void button2\_Click\_1(System::Object^ sender, System::EventArgs^ e){

Teor ^form = gcnew Teor();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button4\_Click\_1(System::Object^ sender, System::EventArgs^ e){

Teor2 ^form = gcnew Teor2();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button6\_Click\_1(System::Object^ sender, System::EventArgs^ e){

ProgressForm ^form = gcnew ProgressForm();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button7\_Click\_1(System::Object^ sender, System::EventArgs^ e){}

private: System::Void button8\_Click\_1(System::Object^ sender, System::EventArgs^ e){

AchivmentsForm ^form = gcnew AchivmentsForm();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button2\_Click\_2(System::Object^ sender, System::EventArgs^ e){

Teor ^form = gcnew Teor();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

Achivments::Achivments\_get("2", 2, ConvertToString(Login\_Text));

TestSelect ^form = gcnew TestSelect();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

Teor2 ^form = gcnew Teor2();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button6\_Click(System::Object^ sender, System::EventArgs^ e) {

ProgressForm ^form = gcnew ProgressForm();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button7\_Click(System::Object^ sender, System::EventArgs^ e){

Cabinet^form = gcnew Cabinet();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button8\_Click(System::Object^ sender, System::EventArgs^ e){

AchivmentsForm ^form = gcnew AchivmentsForm();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button10\_Click(System::Object^ sender, System::EventArgs^ e) {

std::exit(0);

}

private: System::Void MainMenuForm\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

};

}

**ProgressForm.h**

#pragma once

#include "AchivmentsForm.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

/// <summary>

/// Сводка для ProgressForm

/// </summary>

public ref class ProgressForm : public System::Windows::Forms::Form

{

public:

ProgressForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~ProgressForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Label^ label1;

protected:

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button32;

private: System::Windows::Forms::Button^ button33;

private: System::Windows::Forms::Button^ button34;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Button^ button6;

private: System::Windows::Forms::Button^ button7;

private: System::Windows::Forms::Button^ button8;

private: System::Windows::Forms::Button^ button9;

private: System::Windows::Forms::Button^ button10;

private: System::Windows::Forms::Button^ button11;

private: System::Windows::Forms::Button^ button12;

private: System::Windows::Forms::Button^ button13;

private: System::Windows::Forms::Button^ button14;

private: System::Windows::Forms::Button^ button15;

private: System::Windows::Forms::Button^ button16;

private: System::Windows::Forms::Button^ button17;

private: System::Windows::Forms::Button^ button18;

private: System::Windows::Forms::Button^ button19;

private: System::Windows::Forms::Button^ button20;

private: System::Windows::Forms::Button^ button21;

private: System::Windows::Forms::Button^ button22;

private: System::Windows::Forms::Button^ button23;

private: System::Windows::Forms::Button^ button24;

private: System::Windows::Forms::Button^ button25;

private: System::Windows::Forms::Button^ button26;

private: System::Windows::Forms::Button^ button27;

private: System::Windows::Forms::Button^ button28;

private: System::Windows::Forms::Button^ button29;

private: System::Windows::Forms::Button^ button30;

private: System::Windows::Forms::Button^ button31;

private: System::Windows::Forms::Button^ button35;

private: System::Windows::Forms::Button^ button36;

private: System::Windows::Forms::Panel^ panel1;

private: System::Windows::Forms::PictureBox^ pictureBox1;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->label1 = (gcnew System::Windows::Forms::Label());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button32 = (gcnew System::Windows::Forms::Button());

this->button33 = (gcnew System::Windows::Forms::Button());

this->button34 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->button6 = (gcnew System::Windows::Forms::Button());

this->button7 = (gcnew System::Windows::Forms::Button());

this->button8 = (gcnew System::Windows::Forms::Button());

this->button9 = (gcnew System::Windows::Forms::Button());

this->button10 = (gcnew System::Windows::Forms::Button());

this->button11 = (gcnew System::Windows::Forms::Button());

this->button12 = (gcnew System::Windows::Forms::Button());

this->button13 = (gcnew System::Windows::Forms::Button());

this->button14 = (gcnew System::Windows::Forms::Button());

this->button15 = (gcnew System::Windows::Forms::Button());

this->button16 = (gcnew System::Windows::Forms::Button());

this->button17 = (gcnew System::Windows::Forms::Button());

this->button18 = (gcnew System::Windows::Forms::Button());

this->button19 = (gcnew System::Windows::Forms::Button());

this->button20 = (gcnew System::Windows::Forms::Button());

this->button21 = (gcnew System::Windows::Forms::Button());

this->button22 = (gcnew System::Windows::Forms::Button());

this->button23 = (gcnew System::Windows::Forms::Button());

this->button24 = (gcnew System::Windows::Forms::Button());

this->button25 = (gcnew System::Windows::Forms::Button());

this->button26 = (gcnew System::Windows::Forms::Button());

this->button27 = (gcnew System::Windows::Forms::Button());

this->button28 = (gcnew System::Windows::Forms::Button());

this->button29 = (gcnew System::Windows::Forms::Button());

this->button30 = (gcnew System::Windows::Forms::Button());

this->button31 = (gcnew System::Windows::Forms::Button());

this->button35 = (gcnew System::Windows::Forms::Button());

this->button36 = (gcnew System::Windows::Forms::Button());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->panel1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

this->SuspendLayout();

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(75, 31);

this->label1->Margin = System::Windows::Forms::Padding(4, 0, 4, 0);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(350, 37);

this->label1->TabIndex = 0;

this->label1->Text = L"Прогресс выполнения";

//

// button1

//

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button1->Location = System::Drawing::Point(64, 90);

this->button1->Margin = System::Windows::Forms::Padding(4);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(107, 98);

this->button1->TabIndex = 1;

this->button1->Text = L"1";

this->button1->UseVisualStyleBackColor = true;

//

// button32

//

this->button32->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button32->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button32->Location = System::Drawing::Point(597, 22);

this->button32->Margin = System::Windows::Forms::Padding(4);

this->button32->Name = L"button32";

this->button32->Size = System::Drawing::Size(240, 54);

this->button32->TabIndex = 32;

this->button32->Text = L"Январь";

this->button32->UseVisualStyleBackColor = true;

//

// button33

//

this->button33->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button33->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button33->Location = System::Drawing::Point(464, 22);

this->button33->Margin = System::Windows::Forms::Padding(4);

this->button33->Name = L"button33";

this->button33->Size = System::Drawing::Size(107, 54);

this->button33->TabIndex = 33;

this->button33->Text = L"<";

this->button33->UseVisualStyleBackColor = true;

this->button33->Click += gcnew System::EventHandler(this, &ProgressForm::button33\_Click);

//

// button34

//

this->button34->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button34->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button34->Location = System::Drawing::Point(864, 22);

this->button34->Margin = System::Windows::Forms::Padding(4);

this->button34->Name = L"button34";

this->button34->Size = System::Drawing::Size(107, 54);

this->button34->TabIndex = 34;

this->button34->Text = L">";

this->button34->UseVisualStyleBackColor = true;

this->button34->Click += gcnew System::EventHandler(this, &ProgressForm::button34\_Click);

//

// button2

//

this->button2->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button2->Location = System::Drawing::Point(197, 90);

this->button2->Margin = System::Windows::Forms::Padding(4);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(107, 98);

this->button2->TabIndex = 35;

this->button2->Text = L"2";

this->button2->UseVisualStyleBackColor = true;

//

// button3

//

this->button3->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button3->Location = System::Drawing::Point(331, 90);

this->button3->Margin = System::Windows::Forms::Padding(4);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(107, 98);

this->button3->TabIndex = 36;

this->button3->Text = L"3";

this->button3->UseVisualStyleBackColor = true;

//

// button4

//

this->button4->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button4->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button4->Location = System::Drawing::Point(464, 90);

this->button4->Margin = System::Windows::Forms::Padding(4);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(107, 98);

this->button4->TabIndex = 37;

this->button4->Text = L"4";

this->button4->UseVisualStyleBackColor = true;

//

// button5

//

this->button5->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button5->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button5->Location = System::Drawing::Point(597, 90);

this->button5->Margin = System::Windows::Forms::Padding(4);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(107, 98);

this->button5->TabIndex = 38;

this->button5->Text = L"5";

this->button5->UseVisualStyleBackColor = true;

//

// button6

//

this->button6->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button6->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button6->Location = System::Drawing::Point(731, 90);

this->button6->Margin = System::Windows::Forms::Padding(4);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(107, 98);

this->button6->TabIndex = 39;

this->button6->Text = L"6";

this->button6->UseVisualStyleBackColor = true;

//

// button7

//

this->button7->BackColor = System::Drawing::SystemColors::Control;

this->button7->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button7->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button7->Location = System::Drawing::Point(864, 90);

this->button7->Margin = System::Windows::Forms::Padding(4);

this->button7->Name = L"button7";

this->button7->Size = System::Drawing::Size(107, 98);

this->button7->TabIndex = 40;

this->button7->Text = L"7";

this->button7->UseVisualStyleBackColor = false;

//

// button8

//

this->button8->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button8->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button8->Location = System::Drawing::Point(64, 208);

this->button8->Margin = System::Windows::Forms::Padding(4);

this->button8->Name = L"button8";

this->button8->Size = System::Drawing::Size(107, 98);

this->button8->TabIndex = 47;

this->button8->Text = L"8";

this->button8->UseVisualStyleBackColor = true;

//

// button9

//

this->button9->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button9->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button9->Location = System::Drawing::Point(197, 208);

this->button9->Margin = System::Windows::Forms::Padding(4);

this->button9->Name = L"button9";

this->button9->Size = System::Drawing::Size(107, 98);

this->button9->TabIndex = 46;

this->button9->Text = L"9";

this->button9->UseVisualStyleBackColor = true;

//

// button10

//

this->button10->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button10->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button10->Location = System::Drawing::Point(331, 208);

this->button10->Margin = System::Windows::Forms::Padding(4);

this->button10->Name = L"button10";

this->button10->Size = System::Drawing::Size(107, 98);

this->button10->TabIndex = 45;

this->button10->Text = L"10";

this->button10->UseVisualStyleBackColor = true;

//

// button11

//

this->button11->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button11->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button11->Location = System::Drawing::Point(464, 208);

this->button11->Margin = System::Windows::Forms::Padding(4);

this->button11->Name = L"button11";

this->button11->Size = System::Drawing::Size(107, 98);

this->button11->TabIndex = 44;

this->button11->Text = L"11";

this->button11->UseVisualStyleBackColor = true;

//

// button12

//

this->button12->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button12->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button12->Location = System::Drawing::Point(597, 208);

this->button12->Margin = System::Windows::Forms::Padding(4);

this->button12->Name = L"button12";

this->button12->Size = System::Drawing::Size(107, 98);

this->button12->TabIndex = 43;

this->button12->Text = L"12";

this->button12->UseVisualStyleBackColor = true;

//

// button13

//

this->button13->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button13->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button13->Location = System::Drawing::Point(731, 208);

this->button13->Margin = System::Windows::Forms::Padding(4);

this->button13->Name = L"button13";

this->button13->Size = System::Drawing::Size(107, 98);

this->button13->TabIndex = 42;

this->button13->Text = L"13";

this->button13->UseVisualStyleBackColor = true;

//

// button14

//

this->button14->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button14->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button14->Location = System::Drawing::Point(864, 208);

this->button14->Margin = System::Windows::Forms::Padding(4);

this->button14->Name = L"button14";

this->button14->Size = System::Drawing::Size(107, 98);

this->button14->TabIndex = 41;

this->button14->Text = L"14";

this->button14->UseVisualStyleBackColor = true;

//

// button15

//

this->button15->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button15->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button15->Location = System::Drawing::Point(64, 330);

this->button15->Margin = System::Windows::Forms::Padding(4);

this->button15->Name = L"button15";

this->button15->Size = System::Drawing::Size(107, 98);

this->button15->TabIndex = 54;

this->button15->Text = L"15";

this->button15->UseVisualStyleBackColor = true;

//

// button16

//

this->button16->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button16->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button16->Location = System::Drawing::Point(197, 330);

this->button16->Margin = System::Windows::Forms::Padding(4);

this->button16->Name = L"button16";

this->button16->Size = System::Drawing::Size(107, 98);

this->button16->TabIndex = 53;

this->button16->Text = L"16";

this->button16->UseVisualStyleBackColor = true;

//

// button17

//

this->button17->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button17->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button17->Location = System::Drawing::Point(331, 330);

this->button17->Margin = System::Windows::Forms::Padding(4);

this->button17->Name = L"button17";

this->button17->Size = System::Drawing::Size(107, 98);

this->button17->TabIndex = 52;

this->button17->Text = L"17";

this->button17->UseVisualStyleBackColor = true;

//

// button18

//

this->button18->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button18->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button18->Location = System::Drawing::Point(464, 330);

this->button18->Margin = System::Windows::Forms::Padding(4);

this->button18->Name = L"button18";

this->button18->Size = System::Drawing::Size(107, 98);

this->button18->TabIndex = 51;

this->button18->Text = L"18";

this->button18->UseVisualStyleBackColor = true;

//

// button19

//

this->button19->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button19->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button19->Location = System::Drawing::Point(597, 330);

this->button19->Margin = System::Windows::Forms::Padding(4);

this->button19->Name = L"button19";

this->button19->Size = System::Drawing::Size(107, 98);

this->button19->TabIndex = 50;

this->button19->Text = L"19";

this->button19->UseVisualStyleBackColor = true;

//

// button20

//

this->button20->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button20->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button20->Location = System::Drawing::Point(731, 330);

this->button20->Margin = System::Windows::Forms::Padding(4);

this->button20->Name = L"button20";

this->button20->Size = System::Drawing::Size(107, 98);

this->button20->TabIndex = 49;

this->button20->Text = L"20";

this->button20->UseVisualStyleBackColor = true;

//

// button21

//

this->button21->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button21->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button21->Location = System::Drawing::Point(864, 330);

this->button21->Margin = System::Windows::Forms::Padding(4);

this->button21->Name = L"button21";

this->button21->Size = System::Drawing::Size(107, 98);

this->button21->TabIndex = 48;

this->button21->Text = L"21";

this->button21->UseVisualStyleBackColor = true;

//

// button22

//

this->button22->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button22->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button22->Location = System::Drawing::Point(64, 452);

this->button22->Margin = System::Windows::Forms::Padding(4);

this->button22->Name = L"button22";

this->button22->Size = System::Drawing::Size(107, 98);

this->button22->TabIndex = 61;

this->button22->Text = L"22";

this->button22->UseVisualStyleBackColor = true;

//

// button23

//

this->button23->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button23->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button23->Location = System::Drawing::Point(197, 452);

this->button23->Margin = System::Windows::Forms::Padding(4);

this->button23->Name = L"button23";

this->button23->Size = System::Drawing::Size(107, 98);

this->button23->TabIndex = 60;

this->button23->Text = L"23";

this->button23->UseVisualStyleBackColor = true;

//

// button24

//

this->button24->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button24->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button24->Location = System::Drawing::Point(331, 452);

this->button24->Margin = System::Windows::Forms::Padding(4);

this->button24->Name = L"button24";

this->button24->Size = System::Drawing::Size(107, 98);

this->button24->TabIndex = 59;

this->button24->Text = L"24";

this->button24->UseVisualStyleBackColor = true;

//

// button25

//

this->button25->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button25->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button25->Location = System::Drawing::Point(464, 452);

this->button25->Margin = System::Windows::Forms::Padding(4);

this->button25->Name = L"button25";

this->button25->Size = System::Drawing::Size(107, 98);

this->button25->TabIndex = 58;

this->button25->Text = L"25";

this->button25->UseVisualStyleBackColor = true;

//

// button26

//

this->button26->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button26->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button26->Location = System::Drawing::Point(597, 452);

this->button26->Margin = System::Windows::Forms::Padding(4);

this->button26->Name = L"button26";

this->button26->Size = System::Drawing::Size(107, 98);

this->button26->TabIndex = 57;

this->button26->Text = L"26";

this->button26->UseVisualStyleBackColor = true;

//

// button27

//

this->button27->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button27->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button27->Location = System::Drawing::Point(731, 452);

this->button27->Margin = System::Windows::Forms::Padding(4);

this->button27->Name = L"button27";

this->button27->Size = System::Drawing::Size(107, 98);

this->button27->TabIndex = 56;

this->button27->Text = L"27";

this->button27->UseVisualStyleBackColor = true;

//

// button28

//

this->button28->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button28->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button28->Location = System::Drawing::Point(864, 452);

this->button28->Margin = System::Windows::Forms::Padding(4);

this->button28->Name = L"button28";

this->button28->Size = System::Drawing::Size(107, 98);

this->button28->TabIndex = 55;

this->button28->Text = L"28";

this->button28->UseVisualStyleBackColor = true;

//

// button29

//

this->button29->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button29->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button29->Location = System::Drawing::Point(64, 577);

this->button29->Margin = System::Windows::Forms::Padding(4);

this->button29->Name = L"button29";

this->button29->Size = System::Drawing::Size(107, 98);

this->button29->TabIndex = 68;

this->button29->Text = L"29";

this->button29->UseVisualStyleBackColor = true;

//

// button30

//

this->button30->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button30->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button30->Location = System::Drawing::Point(197, 577);

this->button30->Margin = System::Windows::Forms::Padding(4);

this->button30->Name = L"button30";

this->button30->Size = System::Drawing::Size(107, 98);

this->button30->TabIndex = 67;

this->button30->Text = L"30";

this->button30->UseVisualStyleBackColor = true;

//

// button31

//

this->button31->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button31->Font = (gcnew System::Drawing::Font(L"Century Gothic", 27.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button31->Location = System::Drawing::Point(331, 577);

this->button31->Margin = System::Windows::Forms::Padding(4);

this->button31->Name = L"button31";

this->button31->Size = System::Drawing::Size(107, 98);

this->button31->TabIndex = 66;

this->button31->Text = L"31";

this->button31->UseVisualStyleBackColor = true;

//

// button35

//

this->button35->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button35->Font = (gcnew System::Drawing::Font(L"Century Gothic", 18, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button35->Location = System::Drawing::Point(464, 577);

this->button35->Margin = System::Windows::Forms::Padding(4);

this->button35->Name = L"button35";

this->button35->Size = System::Drawing::Size(507, 98);

this->button35->TabIndex = 69;

this->button35->Text = L"Достижения";

this->button35->UseVisualStyleBackColor = true;

this->button35->Click += gcnew System::EventHandler(this, &ProgressForm::button35\_Click);

//

// button36

//

this->button36->Location = System::Drawing::Point(979, 22);

this->button36->Margin = System::Windows::Forms::Padding(4);

this->button36->Name = L"button36";

this->button36->Size = System::Drawing::Size(33, 54);

this->button36->TabIndex = 70;

this->button36->Text = L"X";

this->button36->UseVisualStyleBackColor = true;

this->button36->Visible = false;

this->button36->Click += gcnew System::EventHandler(this, &ProgressForm::button36\_Click);

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->pictureBox1);

this->panel1->Controls->Add(this->button36);

this->panel1->Controls->Add(this->button35);

this->panel1->Controls->Add(this->button29);

this->panel1->Controls->Add(this->button30);

this->panel1->Controls->Add(this->button31);

this->panel1->Controls->Add(this->button22);

this->panel1->Controls->Add(this->button23);

this->panel1->Controls->Add(this->button24);

this->panel1->Controls->Add(this->button25);

this->panel1->Controls->Add(this->button26);

this->panel1->Controls->Add(this->button27);

this->panel1->Controls->Add(this->button28);

this->panel1->Controls->Add(this->button15);

this->panel1->Controls->Add(this->button16);

this->panel1->Controls->Add(this->button17);

this->panel1->Controls->Add(this->button18);

this->panel1->Controls->Add(this->button19);

this->panel1->Controls->Add(this->button20);

this->panel1->Controls->Add(this->button21);

this->panel1->Controls->Add(this->button8);

this->panel1->Controls->Add(this->button9);

this->panel1->Controls->Add(this->button10);

this->panel1->Controls->Add(this->button11);

this->panel1->Controls->Add(this->button12);

this->panel1->Controls->Add(this->button13);

this->panel1->Controls->Add(this->button14);

this->panel1->Controls->Add(this->button7);

this->panel1->Controls->Add(this->button6);

this->panel1->Controls->Add(this->button5);

this->panel1->Controls->Add(this->button4);

this->panel1->Controls->Add(this->button3);

this->panel1->Controls->Add(this->button2);

this->panel1->Controls->Add(this->button34);

this->panel1->Controls->Add(this->button33);

this->panel1->Controls->Add(this->button32);

this->panel1->Controls->Add(this->button1);

this->panel1->Controls->Add(this->label1);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Margin = System::Windows::Forms::Padding(4);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(1045, 689);

this->panel1->TabIndex = 71;

//

// pictureBox1

//

this->pictureBox1->Location = System::Drawing::Point(978, 94);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(61, 89);

this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pictureBox1->TabIndex = 71;

this->pictureBox1->TabStop = false;

this->pictureBox1->Click += gcnew System::EventHandler(this, &ProgressForm::pictureBox1\_Click);

//

// ProgressForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(1045, 690);

this->ControlBox = false;

this->Controls->Add(this->panel1);

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"ProgressForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Календарь";

this->Load += gcnew System::EventHandler(this, &ProgressForm::ProgressForm\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &ProgressForm::ProgressForm\_VisibleChanged);

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

this->ResumeLayout(false);

}

#pragma endregion

int Start\_DayIndex = CalendarGlobal[0];

int End\_DayIndex = CalendarGlobal[0];

int Month = 1;

int ChosenMounth = CalendarGlobal[0];

void ApplyData(int StartDayIndex, int EndDayIndex, int DayIndex, Button^BT){

if (DayIndex >= StartDayIndex && DayIndex<= EndDayIndex)

BT->BackColor = System::Drawing::Color::Red;

}

void WhiteOut(){

button1->BackColor = System::Drawing::Color::White;

button2->BackColor = System::Drawing::Color::White;

button3->BackColor = System::Drawing::Color::White;

button4->BackColor = System::Drawing::Color::White;

button5->BackColor = System::Drawing::Color::White;

button6->BackColor = System::Drawing::Color::White;

button7->BackColor = System::Drawing::Color::White;

button8->BackColor = System::Drawing::Color::White;

button9->BackColor = System::Drawing::Color::White;

button10->BackColor = System::Drawing::Color::White;

button11->BackColor = System::Drawing::Color::White;

button12->BackColor = System::Drawing::Color::White;

button13->BackColor = System::Drawing::Color::White;

button14->BackColor = System::Drawing::Color::White;

button15->BackColor = System::Drawing::Color::White;

button16->BackColor = System::Drawing::Color::White;

button17->BackColor = System::Drawing::Color::White;

button18->BackColor = System::Drawing::Color::White;

button19->BackColor = System::Drawing::Color::White;

button20->BackColor = System::Drawing::Color::White;

button21->BackColor = System::Drawing::Color::White;

button22->BackColor = System::Drawing::Color::White;

button23->BackColor = System::Drawing::Color::White;

button24->BackColor = System::Drawing::Color::White;

button25->BackColor = System::Drawing::Color::White;

button26->BackColor = System::Drawing::Color::White;

button27->BackColor = System::Drawing::Color::White;

button28->BackColor = System::Drawing::Color::White;

button29->BackColor = System::Drawing::Color::White;

button30->BackColor = System::Drawing::Color::White;

button31->BackColor = System::Drawing::Color::White;}

void ApplyDataBlock(){

ApplyData(Start\_DayIndex, End\_DayIndex, 1, button1);

ApplyData(Start\_DayIndex, End\_DayIndex, 2, button2);

ApplyData(Start\_DayIndex, End\_DayIndex, 3, button3);

ApplyData(Start\_DayIndex, End\_DayIndex, 4, button4);

ApplyData(Start\_DayIndex, End\_DayIndex, 5, button5);

ApplyData(Start\_DayIndex, End\_DayIndex, 6, button6);

ApplyData(Start\_DayIndex, End\_DayIndex, 7, button7);

ApplyData(Start\_DayIndex, End\_DayIndex, 8, button8);

ApplyData(Start\_DayIndex, End\_DayIndex, 9, button9);

ApplyData(Start\_DayIndex, End\_DayIndex, 10, button10);

ApplyData(Start\_DayIndex, End\_DayIndex, 11, button11);

ApplyData(Start\_DayIndex, End\_DayIndex, 12, button12);

ApplyData(Start\_DayIndex, End\_DayIndex, 13, button13);

ApplyData(Start\_DayIndex, End\_DayIndex, 14, button14);

ApplyData(Start\_DayIndex, End\_DayIndex, 15, button15);

ApplyData(Start\_DayIndex, End\_DayIndex, 16, button16);

ApplyData(Start\_DayIndex, End\_DayIndex, 17, button17);

ApplyData(Start\_DayIndex, End\_DayIndex, 18, button18);

ApplyData(Start\_DayIndex, End\_DayIndex, 19, button19);

ApplyData(Start\_DayIndex, End\_DayIndex, 20, button20);

ApplyData(Start\_DayIndex, End\_DayIndex, 21, button21);

ApplyData(Start\_DayIndex, End\_DayIndex, 22, button22);

ApplyData(Start\_DayIndex, End\_DayIndex, 23, button23);

ApplyData(Start\_DayIndex, End\_DayIndex, 24, button24);

ApplyData(Start\_DayIndex, End\_DayIndex, 25, button25);

ApplyData(Start\_DayIndex, End\_DayIndex, 26, button26);

ApplyData(Start\_DayIndex, End\_DayIndex, 27, button27);

ApplyData(Start\_DayIndex, End\_DayIndex, 28, button28);

ApplyData(Start\_DayIndex, End\_DayIndex, 29, button29);

ApplyData(Start\_DayIndex, End\_DayIndex, 30, button30);

ApplyData(Start\_DayIndex, End\_DayIndex, 31, button31);

}

void Month2000(){

button29->BackColor = System::Drawing::SystemColors::Control;

if (Month == 1){

button32->Text = "Январь";

button29->Visible = 1;

button30->Visible = 1;

button31->Visible = 1;

if (ChosenMounth == 1)

ApplyDataBlock();

}

if (Month == 2){

button32->Text = "Февраль";

button30->Visible = 0;

button31->Visible = 0;

button29->BackColor = System::Drawing::Color::Yellow;

if (ChosenMounth == 2)

ApplyDataBlock();

}

if (Month == 3){

button32->Text = "Март";

button29->Visible = 1;

button30->Visible = 1;

button31->Visible = 1;

if (ChosenMounth == 3)

ApplyDataBlock();

}

if (Month == 4) {

button32->Text = "Апрель";

button31->Visible = 0;

if (ChosenMounth == 4)

ApplyDataBlock();

}

if (Month == 5) {

button32->Text = "Май";

button31->Visible = 1;

if (ChosenMounth == 5)

ApplyDataBlock();

}

if (Month == 6) {

button32->Text = "Июнь";

button31->Visible = 0;

if (ChosenMounth == 6)

ApplyDataBlock();

}

if (Month == 7) {

button32->Text = "Июль";

button31->Visible = 1;

if (ChosenMounth == 7)

ApplyDataBlock();

}

if (Month == 8) {

button32->Text = "Август";

button31->Visible = 1;

if (ChosenMounth == 8)

ApplyDataBlock();

}

if (Month == 9) {

button32->Text = "Сентябрь";

button31->Visible = 0;

if (ChosenMounth == 9)

ApplyDataBlock();

}

if (Month == 10) {

button32->Text = "Октябрь";

button31->Visible = 1;

if (ChosenMounth == 10)

ApplyDataBlock();

}

if (Month == 11) {

button32->Text = "Ноябрь";

button31->Visible = 0;

if (ChosenMounth == 11)

ApplyDataBlock();

}

if (Month == 12){

button32->Text = "Декабрь";

button31->Visible = 1;

if (ChosenMounth == 12)

ApplyDataBlock();

}

}

private: System::Void button34\_Click(System::Object^ sender, System::EventArgs^ e){

Achivments::Achivments\_get("4", 2, ConvertToString(Login\_Text));

WhiteOut();

if (Month == 12)

Month = 1;

else

Month = Month + 1;

Month2000();

}

private: System::Void button33\_Click(System::Object^ sender, System::EventArgs^ e){

Achivments::Achivments\_get("4", 3, ConvertToString(Login\_Text));

WhiteOut();

if (Month == 1)

Month = 12;

else

Month = Month - 1;

Month2000();

}

private: System::Void button35\_Click(System::Object^ sender, System::EventArgs^ e) {

AchivmentsForm ^form = gcnew AchivmentsForm();

form->Owner = this;

form->Show();

this->Hide();

}

private: System::Void button36\_Click(System::Object^ sender, System::EventArgs^ e) {

Owner->Show();

this->Hide();

}

private: System::Void toolStripButton1\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void ProgressForm\_Load(System::Object^ sender, System::EventArgs^ e){

Achivments::Achivments\_get("4", 1, ConvertToString(Login\_Text));

pictureBox1->Image = Image::FromFile(Application::StartupPath + "\\UI\\Arrow.png");

}

private: System::Void ProgressForm\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

private: System::Void pictureBox1\_Click(System::Object^ sender, System::EventArgs^ e){

Owner->Show();

this->Hide();

}

};

}

**Settings.h**

#pragma once

#include "VAR.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace Data::OleDb;

using namespace BrokenPiano;

using namespace std;

/// <summary>

/// Сводка для Settings

/// </summary>

public ref class Settings : public System::Windows::Forms::Form

{

public:

Settings(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Settings()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button5;

protected:

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Panel^ panel1;

private: System::Windows::Forms::Button^ button6;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button5 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button1 = (gcnew System::Windows::Forms::Button());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->button6 = (gcnew System::Windows::Forms::Button());

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// button5

//

this->button5->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button5->Location = System::Drawing::Point(12, 176);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(210, 50);

this->button5->TabIndex = 10;

this->button5->Text = L"1920x1080";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &Settings::button5\_Click);

//

// button4

//

this->button4->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button4->Location = System::Drawing::Point(12, 120);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(210, 50);

this->button4->TabIndex = 9;

this->button4->Text = L"1366x768";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &Settings::button4\_Click);

//

// button3

//

this->button3->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button3->Location = System::Drawing::Point(12, 64);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(210, 50);

this->button3->TabIndex = 8;

this->button3->Text = L"1024x768";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &Settings::button3\_Click);

//

// button2

//

this->button2->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button2->Location = System::Drawing::Point(12, 8);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(210, 50);

this->button2->TabIndex = 7;

this->button2->Text = L"800x600";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &Settings::button2\_Click);

//

// button1

//

this->button1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button1->Location = System::Drawing::Point(12, 299);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(210, 50);

this->button1->TabIndex = 6;

this->button1->Text = L"ОК";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &Settings::button1\_Click);

//

// panel1

//

this->panel1->Controls->Add(this->button6);

this->panel1->Controls->Add(this->button5);

this->panel1->Controls->Add(this->button4);

this->panel1->Controls->Add(this->button3);

this->panel1->Controls->Add(this->button2);

this->panel1->Controls->Add(this->button1);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(235, 362);

this->panel1->TabIndex = 11;

//

// button6

//

this->button6->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button6->Location = System::Drawing::Point(12, 232);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(210, 50);

this->button6->TabIndex = 11;

this->button6->Text = L"Ваше разрешение";

this->button6->UseVisualStyleBackColor = true;

this->button6->Click += gcnew System::EventHandler(this, &Settings::button6\_Click);

//

// Settings

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 17);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(234, 361);

this->ControlBox = false;

this->Controls->Add(this->panel1);

this->Cursor = System::Windows::Forms::Cursors::Default;

this->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"Settings";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterParent;

this->Text = L"Настройки";

this->Load += gcnew System::EventHandler(this, &Settings::Settings\_Load);

this->panel1->ResumeLayout(false);

this->ResumeLayout(false);

}

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

String ^ T;

switch (size\_stage){

case 1: {

T = "800x600";

break;}

case 2: {

T = "1024x768";

break;}

case 3: {

T = "1366x768";

break;}

case 4: {

T = "1920x1080";

break;}

case 10: {

T = "Своё";

break;}

default:

T = "800x600";

break;}

auto p = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application::StartupPath + "\\resource\\Database1.mdb");

p->Open();

try {

auto c = gcnew OleDbCommand("UPDATE [SETTINGS] SET [Size] = ('" + T + "')");

c->Connection = p;

c->ExecuteNonQuery();

}

catch (System::Runtime::InteropServices::COMException ^ situation) {

MessageBox::Show(situation->Message);

}

p->Close();

Owner->Show();

this->Hide();

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 1;}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 2;}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 3;}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 4;}

private: System::Void button6\_Click(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 10;}

private: System::Void Settings\_Load(System::Object^ sender, System::EventArgs^ e) {

this->ClientSize = System::Drawing::Size(234, 361);}

};

}

**Start.h**

#pragma once

#include <iostream>

#include <fstream>

#include "TitulForm.h"

#include "Login.h"

#include "VAR.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace Data::OleDb;

using namespace std;

/// <summary>

/// Сводка для Start

/// </summary>

public ref class Start : public System::Windows::Forms::Form

{

public:

Start(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Start()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Panel^ panel1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// button1

//

this->button1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button1->Location = System::Drawing::Point(12, 299);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(210, 50);

this->button1->TabIndex = 1;

this->button1->Text = L"ОК";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &Start::button1\_Click);

//

// button2

//

this->button2->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button2->Location = System::Drawing::Point(12, 12);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(210, 50);

this->button2->TabIndex = 2;

this->button2->Text = L"800x600";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &Start::button2\_Click\_1);

//

// button3

//

this->button3->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button3->Location = System::Drawing::Point(12, 68);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(210, 50);

this->button3->TabIndex = 3;

this->button3->Text = L"1024x768";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &Start::button3\_Click);

//

// button4

//

this->button4->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button4->Location = System::Drawing::Point(12, 124);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(210, 50);

this->button4->TabIndex = 4;

this->button4->Text = L"1366x768";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &Start::button4\_Click);

//

// button5

//

this->button5->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button5->Location = System::Drawing::Point(12, 180);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(210, 50);

this->button5->TabIndex = 5;

this->button5->Text = L"1920x1080";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &Start::button5\_Click);

//

// panel1

//

this->panel1->Controls->Add(this->button5);

this->panel1->Controls->Add(this->button4);

this->panel1->Controls->Add(this->button3);

this->panel1->Controls->Add(this->button2);

this->panel1->Controls->Add(this->button1);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(234, 362);

this->panel1->TabIndex = 6;

//

// Start

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 17);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Stretch;

this->ClientSize = System::Drawing::Size(234, 361);

this->ControlBox = false;

this->Controls->Add(this->panel1);

this->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->ForeColor = System::Drawing::SystemColors::ControlText;

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"Start";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Выбор разрешения";

this->Load += gcnew System::EventHandler(this, &Start::Start\_Load);

this->panel1->ResumeLayout(false);

this->ResumeLayout(false);

}

#pragma endregion

private: System::Void Start\_Load(System::Object^ sender, System::EventArgs^ e) {

this->ClientSize = System::Drawing::Size(234, 361);

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

String ^ T;

switch (size\_stage){

case 1: {

T = "800x600";

break;}

case 2: {

T = "1024x768";

break;}

case 3: {

T = "1366x768";

break;}

case 4: {

T = "1920x1080";

break;}

default:

T = "800x600";

break;}

auto p = gcnew OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + Application::StartupPath + "\\resource\\Database1.mdb");

p->Open();

try {

auto c = gcnew OleDbCommand("INSERT INTO [SETTINGS] ([Size]) VALUES ('" + T + "')");

c->Connection = p;

c->ExecuteNonQuery();

}

catch (System::Runtime::InteropServices::COMException ^ situation) {

MessageBox::Show(situation->Message);

}

p->Close();

Login ^form = gcnew Login();

form->Owner = this;

form->Show();

this->Hide();}

private: System::Void button2\_Click\_1(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 1;}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 2; }

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 3;}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

size\_stage = 4;}

};

}

**Teor.h**

#pragma once

#include "VAR.h"

#include "TestsHeaders.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

using namespace Obj\_Space;

/// <summary>

/// Сводка для Teor

/// </summary>

public ref class Teor : public System::Windows::Forms::Form

{

public:

Teor(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Teor()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::ComboBox^ comboBox1;

private: System::Windows::Forms::Panel^ panel1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->comboBox1 = (gcnew System::Windows::Forms::ComboBox());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// button1

//

this->button1->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Top | System::Windows::Forms::AnchorStyles::Right));

this->button1->BackColor = System::Drawing::Color::DarkSeaGreen;

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Popup;

this->button1->Location = System::Drawing::Point(676, 10);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(94, 29);

this->button1->TabIndex = 2;

this->button1->Text = L"Назад";

this->button1->UseVisualStyleBackColor = false;

this->button1->Click += gcnew System::EventHandler(this, &Teor::button1\_Click);

//

// textBox2

//

this->textBox2->Anchor = System::Windows::Forms::AnchorStyles::Left;

this->textBox2->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->textBox2->Location = System::Drawing::Point(10, 45);

this->textBox2->Multiline = true;

this->textBox2->Name = L"textBox2";

this->textBox2->ReadOnly = true;

this->textBox2->Size = System::Drawing::Size(760, 19);

this->textBox2->TabIndex = 0;

this->textBox2->TabStop = false;

this->textBox2->Text = L"Название текста";

this->textBox2->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;

//

// textBox1

//

this->textBox1->Anchor = System::Windows::Forms::AnchorStyles::Left;

this->textBox1->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->textBox1->Location = System::Drawing::Point(10, 70);

this->textBox1->Multiline = true;

this->textBox1->Name = L"textBox1";

this->textBox1->ReadOnly = true;

this->textBox1->ScrollBars = System::Windows::Forms::ScrollBars::Vertical;

this->textBox1->Size = System::Drawing::Size(760, 479);

this->textBox1->TabIndex = 0;

this->textBox1->TabStop = false;

this->textBox1->Text = L"Выберите текст!";

//

// comboBox1

//

this->comboBox1->BackColor = System::Drawing::Color::DarkSeaGreen;

this->comboBox1->DropDownStyle = System::Windows::Forms::ComboBoxStyle::DropDownList;

this->comboBox1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->comboBox1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->comboBox1->FormattingEnabled = true;

this->comboBox1->Items->AddRange(gcnew cli::array< System::Object^ >(6) {

L"ALEXANDER POPOV, THE FIRST INVENTOR OF THE RADIO",

L"CONFERENCES, SYMPOSIUMS, CONGRESSES", L"ENVIROMENTAL PROBLEMS", L"MOSCOW TECHNICAL UNIVERSITY OF COMMUNICATIONS AND INFORMATICS",

L"HISTORICAL REVIEW", L"THE RUSSIAN ACADEMY OF SCIENCES"

});

this->comboBox1->Location = System::Drawing::Point(10, 10);

this->comboBox1->Name = L"comboBox1";

this->comboBox1->Size = System::Drawing::Size(660, 29);

this->comboBox1->TabIndex = 3;

this->comboBox1->SelectedIndexChanged += gcnew System::EventHandler(this, &Teor::comboBox1\_SelectedIndexChanged);

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->comboBox1);

this->panel1->Controls->Add(this->button1);

this->panel1->Controls->Add(this->textBox2);

this->panel1->Controls->Add(this->textBox1);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(784, 560);

this->panel1->TabIndex = 4;

//

// Teor

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 17);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Stretch;

this->ClientSize = System::Drawing::Size(784, 561);

this->ControlBox = false;

this->Controls->Add(this->panel1);

this->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;

this->Name = L"Teor";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Пересказы";

this->Load += gcnew System::EventHandler(this, &Teor::Teor\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &Teor::Teor\_VisibleChanged);

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

this->ResumeLayout(false);

}

#pragma endregion

void Teoria(String^&Text1, String^&Text2, String^&Text3, String^&Text4, String^&Text5, String^&Text6) {

Text2 = " The aim of a conference is to provide a forum in which to evaluate and examine a define phenomenon. Before a conference a preliminary work should be performed. It's necessary to establish an Organizing Committee which determines specific purposes, schedule time and time number of participants tor the conference.\r\n"+

" The Organizing Committee composes a conference program and sends letters of institution to different scientific institutions.The structure of the conference is outlined by the Committee and is known to the participants.People who are going to take part in the conference should send their abstracts to the Committee beforehand.\r\n" +

" On the first day a plenary session is held during which the Chairman makes an opening address and repots the basic goals and procedure of the conference.This session includes 3 or 4 lectures by prominent who survey recent developments in their respective areas.Other reports are heard at the session of working group or workshop which are held in the morning and in the afternoon.Each participant is allowed 15 or 20 minutes for the normal presentation at his paper in working group session and 40 minutes for the lecture in the plenary session.Everyone has a chance to participate in the discussion and exchange of the personal experience.\r\n" +

" Conferences should provide a stimulating ground for the exchange of information and for the generation of new ideas.The materials of the conference a published in scientific institutions, higher educational institutions and libraries in their country and abroad.";

Text5 = " The Academy of Sciences was founded in St. Petersburg according to Peter the Great's order of January 28, 1724.\r\n"+

" It was the time when life itself demanded closer contacts between science and practice.The developing industry, transport, trade and economy in the first place, required a thorough study and rapid development of Russia's natural resources.The transformations, initiated by Peter the Great in the first quarter of the 18th century, in the field of industry, agriculture, trade, state management and culture gave rise to the Russian national culture.\r\n" +

" During his multiple visits abroad Peter the Great acquired the first collections which became later a basis of his famous Cabinet of Curiosities, the forefather of all our museums.Already in 1714, he registered officially the Cabinet of Curiosities and Library.\r\n" +

" It was already at the close of the 17th century when a thought occurred to Peter the Great about the creation of an academy of sciences in Russia.\r\n" +

" Later, he formed his original project comprising an academy of sciences proper and two educational establishments, namely, a university and a grammar school.\r\n" +

" In 1721 the court librarian L.Schumacher was sent on mission abroad to select a scientific staff “for the organization of sciences like in Paris, London, Berlin and other places”.\r\n" +

" The problem of the foundation of the Academy of Sciences was discussed at the meeting of the Senate on the 22nd of January attended by major state figures.\r\n" +

" His Imperial majesty ordered that an academy should be established, in which languages will be studied as well as other sciences and the best arts, and books will be translated.\r\n" +

" The academy was called in a different way during many years : Academy, Academy of Sciences, Academy of Sciences and Arts, Emperial Academy of Sciences, but since December 1991 – the Russian Academy of Sciences.";

Text6 = " The Russian Academy of sciences is the higher scientific institution of Russian Federation uniting the most outstanding scientists of the country.\r\n" +

" The Academy conducts research promoting the advancement of all branches of science in Russia.\r\n" +

" It includes such sections as Physics, and Mathematics, Chemistry(кеместри), Geology and Geography, Biology, Engineering Sciences(, History, Economics, Philosophy and Language) and so on.\r\n" +

" The Academy consists of the following Institutions: Research institutes, laboratories, observatories, branches of the Academy, libraries, museums.\r\n" +

" The Russian Academy of Sciences establishes and promotes scientific contacts with scientific academies and other scientific institutions abroad.\r\n" +

" The Academy trains scientific personnel through the postgraduate courses attached to its institutions and conducts work to advance the qualifications of scientific workers.\r\n" +

" The Russian Academy of Sciences consist of full members(academicians), corresponding members, and foreign members.\r\n" +

" The elections of full members, corresponding members, and foreign members of the Academy are held in accordance with the present Charter and the Regulations governing elections of Russian Academy of Sciences subject to apporoval by the Academy's General Meeting.\r\n" +

" The General Meeting of the Academy defines the basic trends in the Academy's Scientific work.\r\n" +

" Sessions of the General Meeting of the Academy are convened whenever necessary but at least twice a year.\r\n" +

" The Presidium of the Academy consists of the Academy's President, Vise - Presidents, Head Scientific Secretary of the Presidium and Presidium members who are elected from among its full members.\r\n" +

" The Presidium of the academy carries out the decisions of the Academy's activities in between General Meetings.\r\n" +

" A section of the Russian Academy of Sciences is a scientific and organizational center uniting within the Academy scientists of one or several fields of science.\r\n" +

" The highest body of the Section of the Academy is the Genera Meeting of the Section.\r\n" +

" The leading body of the Section between General Meeting is the Section Bureau.\r\n" +

" The principal research bodies of the Academy are its research institutes.\r\n" +

" The Academy exchanges publications with scientific institutions, higher institutions of learning, libraries, museums, publishing houses, scientific societies, and government institutions of other countries.";

Text3 = " The planet Earth is only a small part of the universe, but it's the only place where people can live. Today, our planet is in serious danger. Ozone layer depletion, extinction of animals and plants, air and water pollution, overpopulation are the problems that threaten our life on the Earth. But one of the greatest threats to our planet is global climate change.\r\n" +

" The natural greenhouse effect keeps the planet 33 ºC warmer than it would otherwise be.Over the last 200 years human activities have resulted in increasing emissions of greenhouse gases and therefore the earth's temperature rising.Scientists have projected that if people continue to alter the composition of the atmosphere, average surface temperatures could rise between 2 and 3 ºC globally over the next 100 years.This rate of climatic change would cause significant damage to ecological systems and economies on a global scale.\r\n" +

" The only way to solve the problem is to stop burning fossil fuels and to start using clean renewable energy, like wind or solar power.But such conversion to new energy sources is not an easy task.Different companies continue investing money in new oil, coal and gas exploration, despite the climate threat.And although governments have agreed that the problem is serious they are encouraging these companies with favourable tax rates.\r\n" +

" Every year industrialized countries sign different conventions on reducing carbon dioxide and other emissions, but in the meantime little real action is being taken to address the problem.All over the world people join international organizations, like Greenpeace, WWF and others to make governments face up to their responsibilities and protect our planet's flora and fauna.Nowadays we have the time, the money and even the technology to prevent a disaster.But we should always remember that the longer action is delayed, the more drastic it will need to be.";

Text4 = " The MTUCI is one of the largest academic-research centers aimed at educating students and retraining professionals in the field of telecommunication, radio engineering and economics.\r\n" +

" The history of the University goes back to the Moscow Institute of Electrical Engineering and National Communications1 founded in 1921.\r\n" +

" In 1992 the institute was transformed into the Moscow Technical University of Communications and Informatics(MTUCI).\r\n" +

" The teaching staff of the University consists of many professors, assistant professors and lecturers of whom seventy per cent possess scientific degrees of Doctor of Science or Candidate of Science.\r\n" +

" Among the Russian academic institutions the MTUCI was one of the first to introduce multilevel training providing the probation for Bachelors' and Masters' degrees and Engineers.\r\n" +

" Many under - and postgraduate students, doctorants, trainees2 and learners both full time and correspondence study at the University.The academic structure of the University comprises : \r\n" +

" • educational administration, \r\n" +

" • different faculties, \r\n" +

" • departments, \r\n" +

" • laboratories and study - rooms.\r\n" +

" The educational facilities include : \r\n" +

" • non - graduate upgrade center, \r\n" +

" • distance learning center, \r\n" +

" • computer center, \r\n" +

" • training centers.\r\n" +

" The University has two branches : in Rostov - on - the Don and Nizhny Novgorod.\r\n" +

" Its library contains more than 1,000,000 publications on science and technology.\r\n" +

" The University organizes annual scientific and technological conferences.The results of researches are published in scientific journals, monographs and collected articles.\r\n" +

" A distance learning net in Russia has been developed and started to operate under the supervision of the University in collaboration with the largest communication service operators, such as Rostelecom, MGTS and the leading Russian Universities.\r\n" +

" Moreover, the International distance learning project for the CIS countries has been launched in collaboration with the International Telecommunication Union, the Communications Administration of the CIS countries and their leading higher educational centers.\r\n" +

" The scientific schools founded by famous scientists, such as Kotelnikov V.A., Kharkevich A.A., Levin G.A., Mints A.L, Kliatskin I.G., Shulejkin M.V., Pistolkors A.A., Aizenberg G.Z., Kataev S.I., Goron I.E., Terentjev B.P., etc., ensure the high potential of research activities at the University.\r\n" +

" The University participates in a number of federal projects in the field of telecommunication and informatics, projects of the Russian Foundation for Basic Research and the Ministry of Education.\r\n" +

" Investigations and researches are held in 63 research laboratories(RL) and scientific department groups that are united in the MTUCI research center.\r\n" +

" Turning out highly qualified staff is provided through defending Doctor degrees and postgraduate courses.\r\n" +

" My studies at the MTUCI\r\n" +

" I'm a first year student of the first Technical General Faculty of the MTUCI.I study Fundamental Informatics speciality.Our dean is S.Y.Nekoz.We study 5 days a week and every day we have 3 or 4 couples of lessons.We study such disciplines as Analytical Geometry, Linear Algebra, Informatics, Programming, Physics and so on.After completing a full course of education I'll get a Bachelor's degree.I like studying at the MTUCI because I hope the knowledge I get here will help me find an interesting and well - paid job.";

Text1 = " Today, when powerful radio stations transmit the latest news all over the globe we realize how great is the name of Alexander Popov, the first inventor of the radio.\r\n" +

" One hundred years ago electrical engineering was a new science, and Alexander Popov took great interest in it.He made experiments with electric waves.But his work went on under very hard conditions.He had neither special equipment nor money; however he did not give up and managed to achieve results.\r\n" +

" In 1896 a meeting took place in one of the auditoriums of the Petersburg University.The greatest scientists of the city were present there.Popov made a report and even demonstrated the first radiograms in the world.\r\n" +

" After this the government gave him permission to make his experiments on board of a small ship.And that was all.\r\n" +

" At this time an Italian, Marconi, began making the same kind of experiments.He plagiarized Popov's ideas to make money out of them.He left Italy for London, found protection of rich businessmen and the English government, received a patent for his invention, and organized a commercial company.\r\n" +

" In 1899 Popov could build the first radio station in the world.\r\n" +

" In 1903 he took part in the work of the first International Conference on the Wireless.Marconi was also present there, and tried to prove that he was the inventor of the radio.But one French scientist protested against this.\r\n" +

" In 1904 during the Russian - Japanese war the government realized the importance of the radio.Ships could communicate at long distances only by means of the radio.It was ready to supply Popov with all the necessary equipment but as it was wartime it was impossible to do anything.\r\n" +

" Popov died in 1905. A few years after his death, the Russian Physical Society set up a commission which stated that Popov was the first inventor of the radio.";

}

private: System::Void Teor\_Load(System::Object^ sender, System::EventArgs^ e) {

Achivments::Achivments\_get("0", 1, ConvertToString(Login\_Text)); }

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

Owner->Show();

this->Hide(); }

private: System::Void comboBox1\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

Achivments::Achivments\_get("0", 2, ConvertToString(Login\_Text));

String ^ TextPopov;

String ^ TextConf;

String ^ TextEnvProb;

String ^ TextMTUCI;

String ^ TextHR;

String ^ TextRAS;

Teoria(TextPopov, TextConf, TextEnvProb, TextMTUCI, TextHR, TextRAS);

textBox2->Text = comboBox1->SelectedText;

switch (comboBox1->SelectedIndex) {

case 0: {

textBox1->Text = TextPopov;

break; }

case 1:{

textBox1->Text = TextConf;

break;}

case 2: {

textBox1->Text = TextEnvProb;

break; }

case 3: {

textBox1->Text = TextMTUCI;

break; }

case 4: {

textBox1->Text = TextHR;

break; }

case 5: {

textBox1->Text = TextRAS;

break; }

};

textBox2->Text = Convert::ToString(comboBox1->SelectedItem);

comboBox1->Select(0, 0);

}

private: System::Void Teor\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

};

}

**Teor2.h**

#pragma once

#include "VAR.h"

#include "TestsHeaders.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

using namespace Obj\_Space;

/// <summary>

/// Сводка для Teor2

/// </summary>

public ref class Teor2 : public System::Windows::Forms::Form

{

public:

Teor2(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Teor2()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Panel^ panel1;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(12, 24);

this->textBox1->Multiline = true;

this->textBox1->Name = L"textBox1";

this->textBox1->ReadOnly = true;

this->textBox1->Size = System::Drawing::Size(551, 525);

this->textBox1->TabIndex = 0;

this->textBox1->TabStop = false;

//

// button1

//

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button1->Location = System::Drawing::Point(569, 499);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(203, 50);

this->button1->TabIndex = 1;

this->button1->Text = L"Вернуться";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &Teor2::button1\_Click);

//

// button2

//

this->button2->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button2->Location = System::Drawing::Point(569, 78);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(203, 48);

this->button2->TabIndex = 2;

this->button2->Text = L"Посмотреть примеры резюме";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &Teor2::button2\_Click);

//

// button3

//

this->button3->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button3->Location = System::Drawing::Point(569, 24);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(203, 48);

this->button3->TabIndex = 3;

this->button3->Text = L"Посмотреть структуру резюме";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &Teor2::button3\_Click);

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->button3);

this->panel1->Controls->Add(this->button2);

this->panel1->Controls->Add(this->button1);

this->panel1->Controls->Add(this->textBox1);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(784, 560);

this->panel1->TabIndex = 4;

//

// Teor2

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 17);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(784, 561);

this->ControlBox = false;

this->Controls->Add(this->panel1);

this->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"Teor2";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Резюме";

this->Load += gcnew System::EventHandler(this, &Teor2::Teor2\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &Teor2::Teor2\_VisibleChanged);

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

this->ResumeLayout(false);

}

#pragma endregion

void Resume(bool example, TextBox^T) {

String ^ Text1;

if (example == false) {

Text1 = "1. The author. The headline. The source, year. \r\n" +

"2. THE HEADLINE\r\n" +

"3. The aim of the article is to show the reader that...\r\n" +

"4. The research method used in article is [description/comparison/analysis]\r\n" +

"5. Main Content. (Не копировать, используйте измененные предложения!)\r\n" +

"Примеры, с чего можно начать основную часть:\r\n" +

"The author [states/stressed] that...\r\nThe author start by telling...\r\nThe author point out that...\r\n"

"Объем основной части преполагает 2 абзаца.\r\n" +

"6. The article discribes the period of [вставить нужное] [developmeny/evolution/integration/implementation/using/building/designing] begining with [год] in [место].\r\n " +

"7. The author's conclusion is that...";

}

else {

Text1 = " Sounders R. Banking on ATM by Satellite, Satellite Communications, 1997, p43-45\r\n"+

" BANKING ON ATM BY SATELLITE\r\n"+

" The aim of the article is to show the reader that ATM allows transmitting a phenomenal array of multimedia information in a united structure.\r\n"+

" The research method used in the article is description.\r\n"+

" ATM(Asynchronous Transfer Mode) represent powerful telecommunicatons standard which was originally develop for terrestrial networks and had the ability to convey voice data and video via satellite.Scientist at the Clacksburg, Maryland based Comsat labs reinvnted it to operate on a satellite link since satellite transmissions are subject to noise.For this purpose two interfaces were develop iALA - 2000 for " +

"multinational companies that want to use ATM over thier VSAT networks and ALE - 2000 which targets large telecomms network operators.\r\n"+

" The article describes the period of ATM's reinvention in the 1990s in the USA.\r\n"+

" The author's conclusion is that ATM has many pros and despite it is in precommercical phase, it is expected to be well accepted in the nearest future.";

}

T->Text = Text1;

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

Owner->Show();

this->Hide();

}

private: System::Void Teor2\_Load(System::Object^ sender, System::EventArgs^ e) {

Resume(false, textBox1);

Achivments::Achivments\_get("1", 1, ConvertToString(Login\_Text));

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

Achivments::Achivments\_get("0", 3, ConvertToString(Login\_Text));

Achivments::Achivments\_get("1", 2, ConvertToString(Login\_Text));

Resume(true, textBox1);

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

Resume(false, textBox1);

Achivments::Achivments\_get("1", 3, ConvertToString(Login\_Text));

}

private: System::Void Teor2\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

};

}

**TestSelect.h**

#pragma once

#include "AllTests.h"

#include "VAR.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

/// <summary>

/// Сводка для TestSelect

/// </summary>

public ref class TestSelect : public System::Windows::Forms::Form

{

public:

TestSelect(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~TestSelect()

{

if (components)

{

delete components;

}

}

protected:

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::CheckBox^ checkBox1;

private: System::Windows::Forms::CheckBox^ checkBox2;

private: System::Windows::Forms::CheckBox^ checkBox3;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Timer^ ButtonAction;

private: System::Windows::Forms::Timer^ ButtonUp;

private: System::Windows::Forms::Panel^ panel1;

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->button5 = (gcnew System::Windows::Forms::Button());

this->checkBox1 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox2 = (gcnew System::Windows::Forms::CheckBox());

this->checkBox3 = (gcnew System::Windows::Forms::CheckBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->ButtonAction = (gcnew System::Windows::Forms::Timer(this->components));

this->ButtonUp = (gcnew System::Windows::Forms::Timer(this->components));

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// button5

//

this->button5->BackColor = System::Drawing::Color::Silver;

this->button5->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button5->Location = System::Drawing::Point(711, 32);

this->button5->Margin = System::Windows::Forms::Padding(4);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(60, 60);

this->button5->TabIndex = 4;

this->button5->Text = L"Назад";

this->button5->UseVisualStyleBackColor = false;

this->button5->Click += gcnew System::EventHandler(this, &TestSelect::button5\_Click);

//

// checkBox1

//

this->checkBox1->AutoSize = true;

this->checkBox1->BackColor = System::Drawing::SystemColors::Control;

this->checkBox1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->checkBox1->Location = System::Drawing::Point(97, 164);

this->checkBox1->Name = L"checkBox1";

this->checkBox1->Size = System::Drawing::Size(254, 37);

this->checkBox1->TabIndex = 5;

this->checkBox1->Text = L"Перевод фразы";

this->checkBox1->UseVisualStyleBackColor = false;

//

// checkBox2

//

this->checkBox2->AutoSize = true;

this->checkBox2->BackColor = System::Drawing::SystemColors::Control;

this->checkBox2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->checkBox2->Location = System::Drawing::Point(97, 235);

this->checkBox2->Name = L"checkBox2";

this->checkBox2->Size = System::Drawing::Size(216, 37);

this->checkBox2->TabIndex = 6;

this->checkBox2->Text = L"Ввод вручную";

this->checkBox2->UseVisualStyleBackColor = false;

//

// checkBox3

//

this->checkBox3->AutoSize = true;

this->checkBox3->BackColor = System::Drawing::SystemColors::Control;

this->checkBox3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->checkBox3->Location = System::Drawing::Point(97, 313);

this->checkBox3->Name = L"checkBox3";

this->checkBox3->Size = System::Drawing::Size(216, 37);

this->checkBox3->TabIndex = 7;

this->checkBox3->Text = L"Выбор ответа";

this->checkBox3->UseVisualStyleBackColor = false;

//

// button1

//

this->button1->BackColor = System::Drawing::SystemColors::ActiveBorder;

this->button1->Enabled = false;

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button1->Location = System::Drawing::Point(296, 446);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(244, 69);

this->button1->TabIndex = 11;

this->button1->Text = L"Начать тест";

this->button1->UseVisualStyleBackColor = false;

this->button1->Click += gcnew System::EventHandler(this, &TestSelect::button1\_Click);

//

// ButtonAction

//

this->ButtonAction->Enabled = true;

this->ButtonAction->Tick += gcnew System::EventHandler(this, &TestSelect::ButtonAction\_Tick);

//

// ButtonUp

//

this->ButtonUp->Enabled = true;

this->ButtonUp->Tick += gcnew System::EventHandler(this, &TestSelect::ButtonUp\_Tick);

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->button1);

this->panel1->Controls->Add(this->checkBox3);

this->panel1->Controls->Add(this->checkBox2);

this->panel1->Controls->Add(this->checkBox1);

this->panel1->Controls->Add(this->button5);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(784, 560);

this->panel1->TabIndex = 12;

//

// TestSelect

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 17);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(784, 561);

this->ControlBox = false;

this->Controls->Add(this->panel1);

this->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"TestSelect";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Выбор теста";

this->Load += gcnew System::EventHandler(this, &TestSelect::TestSelect\_Load);

this->VisibleChanged += gcnew System::EventHandler(this, &TestSelect::TestSelect\_VisibleChanged);

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

this->ResumeLayout(false);

}

#pragma endregion

int XS = 244;

int YS = 69;

int XL = 290;

int YL = 446;

int GreyIndex = 120;

int GreenIndex = 240;

void Type\_Select() {

if (checkBox1->Checked == true)

Test\_Types[0] = true;

else

Test\_Types[0] = false;

if (checkBox2->Checked == true)

Test\_Types[1] = true;

else

Test\_Types[1] = false;

if (checkBox3->Checked == true)

Test\_Types[2] = true;

else

Test\_Types[2] = false;

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

Owner->Show();

this->Hide();}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

AllTests ^form = gcnew AllTests();

form->Owner = this;

Type\_Select();

form->Show();this->Hide();

}

private: System::Void TestSelect\_Load(System::Object^ sender, System::EventArgs^ e){

ButtonUp->Enabled = 0;}

private: System::Void ButtonAction\_Tick(System::Object^ sender, System::EventArgs^ e) {

if ((checkBox1->Checked == 1) || (checkBox2->Checked == 1) || (checkBox3->Checked == 1))

ButtonUp->Enabled = 1;

if (checkBox1->Checked == 0 && checkBox2->Checked == 0 && checkBox3->Checked == 0)

button1->Enabled = 0;

}

private: System::Void ButtonUp\_Tick(System::Object^ sender, System::EventArgs^ e) {

if (XS < 450 && XL > 190) {

XS = XS + 10;

button1->Size = System::Drawing::Size(XS, YS);

XL = XL - 5;

button1->Location = System::Drawing::Point(XL, YL); }

if (XS > 300)

button1->Enabled = 1;

if (GreyIndex < 250){

GreyIndex = GreyIndex + 10;

button1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(GreyIndex)), static\_cast<System::Int32>(static\_cast<System::Byte>(GreyIndex)),

static\_cast<System::Int32>(static\_cast<System::Byte>(GreyIndex)));

}

if (GreenIndex > 50){

GreenIndex = GreenIndex - 10;

button1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(GreenIndex)), static\_cast<System::Int32>(static\_cast<System::Byte>(240)),

static\_cast<System::Int32>(static\_cast<System::Byte>(GreenIndex)));

}

if (GreenIndex == 60)

ButtonUp->Enabled = 0;

}

private: System::Void TestSelect\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);

}

};

}

**TitulForm.h**

#pragma once

#include "MainMenuForm.h"

namespace SirotinKopievskii {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace BrokenPiano;

/// <summary>

/// Сводка для TitulForm

/// </summary>

public ref class TitulForm : public System::Windows::Forms::Form

{

public:

TitulForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~TitulForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button4;

protected:

private: System::Windows::Forms::Button^ button3;

internal: System::Windows::Forms::TextBox^ TextBox2;

private:

internal: System::Windows::Forms::TextBox^ TextBox1;

private: System::Windows::Forms::Panel^ panel1;

internal:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(TitulForm::typeid));

this->button4 = (gcnew System::Windows::Forms::Button());

this->button3 = (gcnew System::Windows::Forms::Button());

this->TextBox2 = (gcnew System::Windows::Forms::TextBox());

this->TextBox1 = (gcnew System::Windows::Forms::TextBox());

this->panel1 = (gcnew System::Windows::Forms::Panel());

this->panel1->SuspendLayout();

this->SuspendLayout();

//

// button4

//

this->button4->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button4->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button4->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F));

this->button4->Location = System::Drawing::Point(592, 487);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(177, 31);

this->button4->TabIndex = 25;

this->button4->Text = L"Завершить программу";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &TitulForm::button4\_Click);

//

// button3

//

this->button3->Anchor = System::Windows::Forms::AnchorStyles::None;

this->button3->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button3->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F));

this->button3->Location = System::Drawing::Point(320, 466);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(147, 73);

this->button3->TabIndex = 24;

this->button3->Text = L"Начать работу";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &TitulForm::button3\_Click);

//

// TextBox2

//

this->TextBox2->Anchor = System::Windows::Forms::AnchorStyles::None;

this->TextBox2->BackColor = System::Drawing::SystemColors::Control;

this->TextBox2->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->TextBox2->Cursor = System::Windows::Forms::Cursors::Default;

this->TextBox2->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F));

this->TextBox2->Location = System::Drawing::Point(495, 282);

this->TextBox2->MaxLength = 300;

this->TextBox2->Multiline = true;

this->TextBox2->Name = L"TextBox2";

this->TextBox2->ReadOnly = true;

this->TextBox2->ShortcutsEnabled = false;

this->TextBox2->Size = System::Drawing::Size(265, 139);

this->TextBox2->TabIndex = 21;

this->TextBox2->TabStop = false;

this->TextBox2->Text = L"Выполнили студенты группы БФИ1802\r\nСиротин Н. С.\r\nКопиевский Н. Ю.\r\nНаучный руков"

L"одитель:\r\nдоцент кафедры Информатика\r\nк.п.н. Гуриков С. Р.\r\nдоцент кафедры Иност"

L"ранные языки\r\nк.п.н Кожевникова Т.В.\r\n";

this->TextBox2->TextAlign = System::Windows::Forms::HorizontalAlignment::Right;

//

// TextBox1

//

this->TextBox1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->TextBox1->BackColor = System::Drawing::SystemColors::Control;

this->TextBox1->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->TextBox1->Cursor = System::Windows::Forms::Cursors::Default;

this->TextBox1->Font = (gcnew System::Drawing::Font(L"Century Gothic", 9.75F));

this->TextBox1->Location = System::Drawing::Point(15, 12);

this->TextBox1->MaxLength = 300;

this->TextBox1->Multiline = true;

this->TextBox1->Name = L"TextBox1";

this->TextBox1->ReadOnly = true;

this->TextBox1->ShortcutsEnabled = false;

this->TextBox1->Size = System::Drawing::Size(745, 197);

this->TextBox1->TabIndex = 20;

this->TextBox1->TabStop = false;

this->TextBox1->Text = resources->GetString(L"TextBox1.Text");

this->TextBox1->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;

//

// panel1

//

this->panel1->Anchor = System::Windows::Forms::AnchorStyles::None;

this->panel1->Controls->Add(this->TextBox1);

this->panel1->Controls->Add(this->TextBox2);

this->panel1->Controls->Add(this->button3);

this->panel1->Controls->Add(this->button4);

this->panel1->Location = System::Drawing::Point(0, 0);

this->panel1->Name = L"panel1";

this->panel1->Size = System::Drawing::Size(784, 560);

this->panel1->TabIndex = 26;

//

// TitulForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(784, 561);

this->ControlBox = false;

this->Controls->Add(this->panel1);

this->Name = L"TitulForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::Manual;

this->Text = L"Титульный лист";

this->VisibleChanged += gcnew System::EventHandler(this, &TitulForm::TitulForm\_VisibleChanged);

this->panel1->ResumeLayout(false);

this->panel1->PerformLayout();

this->ResumeLayout(false);

}

#pragma endregion

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

MainMenuForm ^form = gcnew MainMenuForm();

form->Owner = this;

form->Show();

this->Hide(); }

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

std::exit(0);}

private: System::Void TitulForm\_VisibleChanged(System::Object^ sender, System::EventArgs^ e) {

ChangeForm\_Module::Resize(this);}

};

}

**TestHeaders.h**

#pragma once

#include <cmath>

#include <ctime>

#include <cstdlib>

#include <string>

using namespace System;

namespace SpaceForTest {

using namespace System::Windows::Forms;

using namespace std;

public ref class AllTest\_Module

{

public:

static ref class PhraseTranslate\_Module

{

public:

static void Make\_Buttons(String^Text1, String^Text2, String^Text3, String^Text4, String^Text5, String^Text6, Button^button0, Button^button1, Button^button2, Button^button3, Button^button4, Button^button5);

static void AnswerCheck(String^Answer, Button^TB1, Button^TB2, Button^TB3, bool \*right\_ans, int qst\_number);

static void Filling(String^Text, Button^B1, Button^B2, Button^B3);

static void RemoveEl(Button^B);

static void CheckTrackBar(TrackBar^TB, Button^BT, bool \*now\_qst);

static void FullCheck(String^Text, Button^B1, Button^B2, Button^B3);

};

static ref class RadioButton\_Module

{

public:

static void MakeButtons(String^Text1, String^Text2, String^Text3, String^Text4, RadioButton^B1, RadioButton^B2, RadioButton^B3, RadioButton^B4);

};

};

}

namespace Obj\_Space {

using namespace SpaceForTest;

String ^ ConvertToString(string Text);

string ConvertTostring(String ^ Text);

public ref class AllTest

{

public:

static void PTran\_Qst(string \*\*PTran);

static void vvodfun(string \*\*vvod);

static void RBLoad(string \*\*Rbutton);

static void all\_load(bool \*Test\_Types, string \*\*PTran, string \*\*vvod, string \*\*Rbutton, Panel ^ p1, Panel ^ p2, Panel^p3, Panel ^ p7, DataGridView ^ dataGridView1, bool \*now\_qst, Button ^button12, int \*ans\_kol, int \*now\_test, string \*\*all\_ans, int n);

static void trackBar1\_changed(Label^label1, string \*\*PTran, int \*ans\_kol, TrackBar ^ trackBar1, bool \*now\_qst, Button^button2, Button^button3, Button^button4, Button^button5, Button^button6, Button^button7, Button^button8, Button^button9, Button^button10, Button^button11, Button^button12);

static void trackBar2\_changed(Label^label2, Label^label3, TextBox^textBox1, string \*\*vvod, int \*ans\_kol, int val, bool \*now\_qst, Button ^ button12, Button^ button13);

static void trackBar3\_change(Label ^ label4, Label ^ label5, string \*\*Rbutton, int \*ans\_kol, int val, bool \*now\_qst, RadioButton ^ rb1, RadioButton ^ rb2, RadioButton ^ rb3, RadioButton ^ rb4, Button ^ button12, Button ^ button14, GroupBox ^groupBox1);

static void button11\_Press(string \*\*PTran, int \*ans\_kol, TrackBar^trackBar1, bool \*now\_qst, Button^button2, Button^button3, Button^button4, Button^button11, Button ^ button12);

static void Answers\_add(string \*\*PTran, string \*\*vvod, string \*\*Rbutton, int \*now\_test, string \*\*all\_ans, int \*n);

static void button12\_Press(bool \*Test\_Types, int \*now\_test, string \*\*PTran, string \*\*vvod, string \*\*Rbutton, Panel ^ p1, Panel ^ p2, Panel^p3, Panel ^ p7, DataGridView ^ dataGridView1, int \*ans\_kol, bool \*now\_qst, Button^button12, Button^button1, string \*\*all\_ans, int \*n);

static void button13\_Press(Label ^ label3, TextBox ^ textBox1, string \*\*vvod, int \*ans\_kol, int val, bool \*now\_qst, Button^button12, Button^ button13);

static void button14\_Press(Label ^ label5, string \*\*Rbutton, int \*ans\_kol, int val, bool \*now\_qst, RadioButton ^ rb1, RadioButton ^ rb2, RadioButton ^ rb3, RadioButton ^ rb4, Button ^ button12, Button ^ button14, GroupBox ^groupBox1);

static void panel1\_Visible(Label ^ label1, string \*\*PTran, Button ^ button2, Button ^ button3, Button ^ button4, Button ^ button5, Button ^ button6, Button ^ button7, Button ^ button8, Button ^ button9, Button ^ button10);

static void panel2\_Visible(Label^label2, Label ^ label3, TextBox^ textBox1, string \*\*vvod, int val);

static void panel3\_Visible(Label ^ label4, Label^ label5, string \*\*Rbutton, int val, RadioButton ^rb1, RadioButton ^rb2, RadioButton ^rb3, RadioButton ^rb4);

static void RB\_Unsel(RadioButton^RB1, RadioButton^RB2, RadioButton^RB3, RadioButton^RB4);

static String ^ GroupBoxSelected(RadioButton^RB1, RadioButton^RB2, RadioButton^RB3, RadioButton^RB4);

static bool Ch(bool \*mas);

static void P\_Visible(Panel^P1, Panel^P2, Panel^P3);

static void Panel\_Change(Panel^P1, Panel^P2, Panel^P3, Panel^P7, int \*now\_test);

static void test\_check(bool \*Test\_Types, int \*now\_test, string \*\*PTran, string \*\*vvod, string \*\*Rbutton, Panel ^ p1, Panel ^ p2, Panel^p3, Panel ^ p7, DataGridView ^ dataGridView1, string \*\*all\_ans, int n);

static void Output\_arr(string \*\*mas, int n, int m, DataGridView ^ DGV);

};

}

**VAR.h**

#pragma once

#include <string>

extern bool \*Test\_Types;

extern int \*CalendarGlobal;

extern int \*AchivmentCheckGlobal;

extern int size\_stage;

extern std::string Login\_Text;

**TestsSource.cpp**

#include "TestsHeaders.h"

namespace SpaceForTest {

void AllTest\_Module::PhraseTranslate\_Module::Make\_Buttons(String ^ Text1, String^Text2, String^Text3, String^Text4, String^Text5, String^Text6, Button^button0, Button^button1, Button^button2, Button^button3, Button^button4, Button^button5) {

srand(time(NULL));

String^Sel\_Text;

int \*BN = new int[6];

int t = 0;

for (int i = 0; i < 6; i++)

BN[i] = i + 1;

for (int i = 0; i < 6; i++) {

int j = (rand() % 5 + 0);

t = BN[i];

BN[i] = BN[j];

BN[j] = t;

}

for (int i = 0; i < 6; i++) {

switch (BN[i]) {

case 1: {

Sel\_Text = Text1;

break;}

case 2: {

Sel\_Text = Text2;

break;}

case 3: {

Sel\_Text = Text3;

break;}

case 4: {

Sel\_Text = Text4;

break;}

case 5: {

Sel\_Text = Text5;

break; }

case 6: {

Sel\_Text = Text6;

break; }

default:

break;

}

switch (i) {

case 0: {

button0->Text = Sel\_Text;

break;}

case 1: {

button1->Text = Sel\_Text;

break; }

case 2: {

button2->Text = Sel\_Text;

break; }

case 3: {

button3->Text = Sel\_Text;

break; }

case 4: {

button4->Text = Sel\_Text;

break; }

case 5: {

button5->Text = Sel\_Text;

break; }

default:

break;

}

}

}

void AllTest\_Module::PhraseTranslate\_Module::AnswerCheck(String^Answer, Button^TB1, Button^TB2, Button^TB3, bool \*right\_ans, int qst\_number) {

String ^ FullAnswer = (TB1->Text + " " + TB2->Text + " " + TB3->Text);

if (FullAnswer == Answer)

right\_ans[qst\_number] = true;

}

void AllTest\_Module::PhraseTranslate\_Module::Filling(String^Text, Button^B1, Button^B2, Button^B3) {

if (B1->Text == "")

B1->Text = Text;

else if (B2->Text == "")

B2->Text = Text;

else if (B3->Text == "")

B3->Text = Text;

}

void AllTest\_Module::PhraseTranslate\_Module::RemoveEl(Button^B) {

B->Text = "";}

void AllTest\_Module::PhraseTranslate\_Module::FullCheck(String^Text, Button^B1, Button^B2, Button^B3) {

if (B1->Text != "" && B2->Text != "" && B3->Text != "")

MessageBox::Show("Пожалуйста удалите лишний элемент!", "Лишний элемент", MessageBoxButtons::OK, MessageBoxIcon::Warning);

else

AllTest\_Module::PhraseTranslate\_Module::Filling(Text, B1, B2, B3);

}

void AllTest\_Module::PhraseTranslate\_Module::CheckTrackBar(TrackBar^trackBar1, Button^button1, bool \*now\_qst) {

if (now\_qst[trackBar1->Value] == true) {

button1->Enabled = false;

button1->Visible = false;

}

else {

button1->Enabled = true;

button1->Visible = true;

}

}

void AllTest\_Module::RadioButton\_Module::MakeButtons(String^Text1, String^Text2, String^Text3, String^Text4, RadioButton^B1, RadioButton^B2, RadioButton^B3, RadioButton^B4) {

srand(time(NULL));

String^Sel\_Text;

int \*BN = new int[4];

int t = 0;

for (int i = 0; i < 4; i++)

BN[i] = i + 1;

for (int i = 0; i < 4; i++) {

int j = (rand() % 3 + 0);

t = BN[i];

BN[i] = BN[j];

BN[j] = t;

}

for (int i = 0; i < 4; i++) {

switch (BN[i]) {

case 1: {

Sel\_Text = Text1;

break;}

case 2: {

Sel\_Text = Text2;

break;}

case 3: {

Sel\_Text = Text3;

break;}

case 4: {

Sel\_Text = Text4;

break; }

default:

break;}

switch (i){

case 0: {

B1->Text = Sel\_Text;

break;}

case 1: {

B2->Text = Sel\_Text;

break;}

case 2: {

B3->Text = Sel\_Text;

break; }

case 3: {

B4->Text = Sel\_Text;

break;}

default:

break;

}

}

}

}

namespace Obj\_Space {

String ^ ConvertToString(string Text) {

String^Text2 = gcnew String(Text.c\_str());

return Text2;

}

string ConvertTostring(String ^ Text) {

using namespace Runtime::InteropServices;

const char \*chars = (const char\*)(Marshal::StringToHGlobalAnsi(Text).ToPointer());

return chars;

}

void AllTest::PTran\_Qst(string \*\*PTran) {

for (int i = 0; i < 5; i++)

PTran[i] = new string[5];

PTran[3][0] = "Draft regulations of the Academia";

PTran[3][1] = "The elections of full members";

PTran[3][2] = "Provide a stimulating ground";

PTran[3][3] = "Preliminary work performed";

PTran[3][4] = "The highest body of the Section";

PTran[4][0] = "проект положений академии";

PTran[4][1] = "выборы полноправных членов";

PTran[4][2] = "обеспечить стимулирующую почву";

PTran[4][3] = "предварительная работа выполнена";

PTran[4][4] = "главный орган секции";

}

void AllTest::vvodfun(string \*\*vvod) {

for (int i = 0; i < 3; i++)

vvod[i] = new string[5];

vvod[0][0] = "A place of study or training in a special field";

vvod[0][1] = "A systematically organized body of knowledge on a particular subject";

vvod[0][2] = "A person who tekes part in something";

vvod[0][3] = "A meeting where ideas and views on a particular issue can be exchanged";

vvod[0][4] = "The systematic investigation into and study of materials and sources in order to establish facts";

vvod[2][0] = "academy";

vvod[2][1] = "science";

vvod[2][2] = "participant";

vvod[2][3] = "conference";

vvod[2][4] = "research";

}

void AllTest::RBLoad(string \*\*Rbutton) {

for (int i = 0; i < 7; i++)

Rbutton[i] = new string[5];

Rbutton[0][0] = "Satellite";

Rbutton[0][1] = "Curiosities";

Rbutton[0][2] = "Chairman";

Rbutton[0][3] = "Headline";

Rbutton[0][4] = "Committe";

Rbutton[1][0] = "Сателит";

Rbutton[1][1] = "Диковина";

Rbutton[1][2] = "Председатель";

Rbutton[1][3] = "Заголовок";

Rbutton[1][4] = "Комиссия";

Rbutton[2][0] = "Спутник";

Rbutton[2][1] = "Явление";

Rbutton[2][2] = "Человек-стул";

Rbutton[2][3] = "Заключение";

Rbutton[2][4] = "Свершение";

Rbutton[3][0] = "Статья";

Rbutton[3][1] = "Исключение";

Rbutton[3][2] = "Участник";

Rbutton[3][3] = "Исключение";

Rbutton[3][4] = "Коммунист";

Rbutton[4][0] = "Заглавие";

Rbutton[4][1] = "Проклятье";

Rbutton[4][2] = "Коммисия";

Rbutton[4][3] = "Выполнение";

Rbutton[4][4] = "Бюро";

Rbutton[6][0] = "Спутник";

Rbutton[6][1] = "Диковина";

Rbutton[6][2] = "Председатель";

Rbutton[6][3] = "Заголовок";

Rbutton[6][4] = "Комиссия";

}

void AllTest::all\_load(bool \*Test\_Types, string \*\*PTran, string \*\*vvod, string \*\*Rbutton, Panel ^ p1, Panel ^ p2, Panel^p3, Panel ^ p7, DataGridView ^ dataGridView1, bool \*now\_qst, Button ^button12, int \*ans\_kol, int \*now\_test, string \*\*all\_ans, int n) {

for (int j = 0; j < 5; j++) {

now\_qst[j] = false;

}

if (ans\_kol[0] >= 5)

button12->Visible = true;

else

button12->Visible = false;

AllTest::test\_check(Test\_Types, now\_test, PTran, vvod, Rbutton, p1, p2, p3, p7, dataGridView1, all\_ans, n);

AllTest::Panel\_Change(p1, p2, p3, p7, now\_test);

p7->Visible = false;

}

void AllTest::trackBar1\_changed(Label^label1, string \*\*PTran, int \*ans\_kol, TrackBar ^ trackBar1, bool \*now\_qst, Button^button2, Button^button3, Button^button4, Button^button5, Button^button6, Button^button7, Button^button8, Button^button9, Button^button10, Button^button11, Button^button12) {

AllTest\_Module::PhraseTranslate\_Module::CheckTrackBar(trackBar1, button11, now\_qst);

switch (trackBar1->Value){

case 0: {

label1->Text = ConvertToString(PTran[3][0]);

AllTest\_Module::PhraseTranslate\_Module::Make\_Buttons("Проект", "Положений", "Учебный", "Правительства", "План", "Академии", button5, button6, button7, button8, button9, button10);

button2->Text = ConvertToString(PTran[0][0]);

button3->Text = ConvertToString(PTran[1][0]);

button4->Text = ConvertToString(PTran[2][0]);

break;

case 1: {

label1->Text = ConvertToString(PTran[3][1]);

AllTest\_Module::PhraseTranslate\_Module::Make\_Buttons("Электричество", "Полноправных", "Членов", "Полностью", "Выборы", "Подключено", button5, button6, button7, button8, button9, button10);

button2->Text = ConvertToString(PTran[0][1]);

button3->Text = ConvertToString(PTran[1][1]);

button4->Text = ConvertToString(PTran[2][1]);

break;

}

case 2: {

label1->Text = ConvertToString(PTran[3][2]);

AllTest\_Module::PhraseTranslate\_Module::Make\_Buttons("Получить", "Обеспечить", "Почву", "Стимулирующую", "Проводящую", "Продолжать", button5, button6, button7, button8, button9, button10);

button2->Text = ConvertToString(PTran[0][2]);

button3->Text = ConvertToString(PTran[1][2]);

button4->Text = ConvertToString(PTran[2][2]);

break;

}

case 3: {

label1->Text = ConvertToString(PTran[3][3]);

AllTest\_Module::PhraseTranslate\_Module::Make\_Buttons("Принудительная", "Работа", "Выполнена", "Дополнительная", "Отменяется", "Предварительная", button5, button6, button7, button8, button9, button10);

button2->Text = ConvertToString(PTran[0][3]);

button3->Text = ConvertToString(PTran[1][3]);

button4->Text = ConvertToString(PTran[2][3]);

break;

}

case 4: {

label1->Text = ConvertToString(PTran[3][4]);

AllTest\_Module::PhraseTranslate\_Module::Make\_Buttons("Высокий", "Высший", "Главный", "Орган", "Секции", "Начальник", button5, button6, button7, button8, button9, button10);

button2->Text = ConvertToString(PTran[0][4]);

button3->Text = ConvertToString(PTran[1][4]);

button4->Text = ConvertToString(PTran[2][4]);

break;

}

}

default:

break;

}

if (ans\_kol[0] >= 5)

button12->Visible = true;

else

button12->Visible = false;

}

void AllTest::trackBar2\_changed(Label^label2, Label^label3, TextBox^textBox1, string \*\*vvod, int \*ans\_kol, int val, bool \*now\_qst, Button ^ button12, Button^ button13) {

label2->Text = ConvertToString(vvod[0][val]);

if (now\_qst[val] == false) {

label3->Visible = false;

textBox1->Visible = true;

textBox1->Text = "";

button13->Visible = true;

}

else {

label3->Visible = true;

label3->Text = ConvertToString(vvod[1][val]);

textBox1->Visible = false;

button13->Visible = false;

}

if (ans\_kol[0] >= 5)

button12->Visible = true;

else

button12->Visible = false;

}

void AllTest::trackBar3\_change(Label ^ label4, Label ^ label5, string \*\*Rbutton, int \*ans\_kol, int val, bool \*now\_qst, RadioButton ^ rb1, RadioButton ^ rb2, RadioButton ^ rb3, RadioButton ^ rb4, Button ^ button12, Button ^ button14, GroupBox ^groupBox1) {

AllTest\_Module::RadioButton\_Module::MakeButtons(ConvertToString(Rbutton[1][val]), ConvertToString(Rbutton[2][val]), ConvertToString(Rbutton[3][val]), ConvertToString(Rbutton[4][val]), rb1, rb2, rb3, rb4);

label4->Text = ConvertToString(Rbutton[0][val]);

AllTest::RB\_Unsel(rb1, rb2, rb3, rb4);

if (now\_qst[val] == false) {

label5->Visible = false;

groupBox1->Visible = true;

button14->Visible = true;

}

else {

label5->Text = ConvertToString(Rbutton[5][val]);

label5->Visible = true;

groupBox1->Visible = false;

button14->Visible = false;

}

if (ans\_kol[0] >= 5)

button12->Visible = true;

else

button12->Visible = false;

}

void AllTest::button11\_Press(string \*\*PTran, int \*ans\_kol, TrackBar^trackBar1, bool \*now\_qst, Button^button2, Button^button3, Button^button4, Button^button11, Button ^ button12) {

if (button2->Text != "" & button3->Text != "" & button4->Text != "") {

PTran[0][trackBar1->Value] = ConvertTostring(button2->Text);

PTran[1][trackBar1->Value] = ConvertTostring(button3->Text);

PTran[2][trackBar1->Value] = ConvertTostring(button4->Text);

now\_qst[trackBar1->Value] = true;

AllTest\_Module::PhraseTranslate\_Module::CheckTrackBar(trackBar1, button11, now\_qst);

ans\_kol[0]++;

if (ans\_kol[0] >= 5)

button12->Visible = true;

else

button12->Visible = false;

}

else {

MessageBox::Show("Пожалуйста, ответьте на вопрос!", "Ответьте на вопрос", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

}

void AllTest::Answers\_add(string \*\*PTran, string \*\*vvod, string \*\*Rbutton, int \*now\_test, string \*\*all\_ans, int \*n) {

if (now\_test[0] == 0) {

for (int i = 0; i < 5; i++) {

String ^ S = ConvertToString(PTran[0][i]) + " " + ConvertToString(PTran[1][i]) + " " + ConvertToString(PTran[2][i]);

all\_ans[n[0]][i] = ConvertTostring(S->ToLower());

}

n[0]++;

for (int i = 0; i < 5; i++)

all\_ans[n[0]][i] = PTran[4][i];

for (int i = 0; i < 5; i++) {

if (all\_ans[n[0]][i] == all\_ans[n[0] - 1][i])

now\_test[1]++;

}

n[0]++;

}

if (now\_test[0] == 1) {

for (int i = 0; i < 5; i++)

all\_ans[n[0]][i] = vvod[1][i];

n[0]++;

for (int i = 0; i < 5; i++)

all\_ans[n[0]][i] = vvod[2][i];

for (int i = 0; i < 5; i++) {

if (all\_ans[n[0]][i] == all\_ans[n[0] - 1][i])

now\_test[1]++;

}

n[0]++;

}

if (now\_test[0] == 2) {

for (int i = 0; i < 5; i++)

all\_ans[n[0]][i] = Rbutton[5][i];

n[0]++;

for (int i = 0; i < 5; i++)

all\_ans[n[0]][i] = Rbutton[6][i];

for (int i = 0; i < 5; i++) {

if (all\_ans[n[0]][i] == all\_ans[n[0] - 1][i])

now\_test[1]++;

}

n[0]++;

}

}

void AllTest::button12\_Press(bool \*Test\_Types, int \*now\_test, string \*\*PTran, string \*\*vvod, string \*\*Rbutton, Panel ^ p1, Panel ^ p2, Panel^p3, Panel ^ p7, DataGridView ^ dataGridView1, int \*ans\_kol, bool \*now\_qst, Button^button12, Button^button1, string \*\*all\_ans, int \*n) {

for (int i = 0; i < sizeof(now\_qst) + 1; i++)

now\_qst[i] = false;

AllTest::Answers\_add(PTran, vvod, Rbutton, now\_test, all\_ans, n);

now\_test[0]++;

ans\_kol[0] = 0;

AllTest::test\_check(Test\_Types, now\_test, PTran, vvod, Rbutton, p1, p2, p3, p7, dataGridView1, all\_ans, n[0]);

AllTest::Panel\_Change(p1, p2, p3, p7, now\_test);

button12->Visible = false;

if (now\_test[0] >= 3)

button1->Visible = false;

}

void AllTest::button13\_Press(Label ^ label3, TextBox ^ textBox1, string \*\*vvod, int \*ans\_kol, int val, bool \*now\_qst, Button^button12, Button^ button13) {

if (textBox1->Text == "")

MessageBox::Show("Пожалуйста, ответьте на вопрос!", "Ответьте на вопрос", MessageBoxButtons::OK, MessageBoxIcon::Warning);

else {

vvod[1][val] = ConvertTostring((textBox1->Text)->ToLower());

label3->Text = textBox1->Text;

textBox1->Visible = false;

textBox1->Text = "";

label3->Visible = true;

now\_qst[val] = true;

ans\_kol[0]++;

if (ans\_kol[0] >= 5)

button12->Visible = true;

else

button12->Visible = false;

button13->Visible = false;

}

}

void AllTest::button14\_Press(Label ^ label5, string \*\*Rbutton, int \*ans\_kol, int val, bool \*now\_qst, RadioButton ^ rb1, RadioButton ^ rb2, RadioButton ^ rb3, RadioButton ^ rb4, Button ^ button12, Button ^ button14, GroupBox ^groupBox1) {

if (AllTest::GroupBoxSelected(rb1, rb2, rb3, rb4) != "") {

String ^ Sel\_ans = AllTest::GroupBoxSelected(rb1, rb2, rb3, rb4);

Rbutton[5][val] = ConvertTostring(Sel\_ans);

label5->Text = Sel\_ans;

label5->Visible = true;

groupBox1->Visible = false;

delete Sel\_ans;

button14->Visible = false;

now\_qst[val] = true;

ans\_kol[0]++;

if (ans\_kol[0] >= 5)

button12->Visible = true;

else

button12->Visible = false;

}

else

MessageBox::Show("Пожалуйста, ответьте на вопрос!", "Ответьте на вопрос", MessageBoxButtons::OK, MessageBoxIcon::Warning);

}

void AllTest::panel1\_Visible(Label ^ label1, string \*\*PTran, Button ^ button2, Button ^ button3, Button ^ button4, Button ^ button5, Button ^ button6, Button ^ button7, Button ^ button8, Button ^ button9, Button ^ button10) {

AllTest\_Module::PhraseTranslate\_Module::Make\_Buttons("Проект", "Положений", "Учебный", "Правительства", "План", "Академии", button5, button6, button7, button8, button9, button10);

button2->Text = "";

button3->Text = "";

button4->Text = "";

AllTest::PTran\_Qst(PTran);

label1->Text = ConvertToString(PTran[3][0]);

}

void AllTest::panel2\_Visible(Label^label2, Label ^ label3, TextBox^ textBox1, string \*\*vvod, int val) {

AllTest::vvodfun(vvod);

label2->Text = ConvertToString(vvod[0][val]);

label3->Visible = false;

textBox1->Visible = true;

textBox1->Text = "";

}

void AllTest::panel3\_Visible(Label ^ label4, Label^ label5, string \*\*Rbutton, int val, RadioButton ^rb1, RadioButton ^rb2, RadioButton ^rb3, RadioButton ^rb4) {

AllTest::RBLoad(Rbutton);

label4->Text = ConvertToString(Rbutton[0][0]);

label5->Visible = false;

AllTest::RB\_Unsel(rb1, rb2, rb3, rb4);

AllTest\_Module::RadioButton\_Module::MakeButtons(ConvertToString(Rbutton[1][val]), ConvertToString(Rbutton[2][val]), ConvertToString(Rbutton[3][val]), ConvertToString(Rbutton[4][val]), rb1, rb2, rb3, rb4);

}

void AllTest::RB\_Unsel(RadioButton^RB1, RadioButton^RB2, RadioButton^RB3, RadioButton^RB4) {

RB1->Checked = false;

RB2->Checked = false;

RB3->Checked = false;

RB4->Checked = false;

}

String ^ AllTest::GroupBoxSelected(RadioButton^RB1, RadioButton^RB2, RadioButton^RB3, RadioButton^RB4) {

if (RB1->Checked)

return RB1->Text;

else if (RB2->Checked)

return RB2->Text;

else if (RB3->Checked)

return RB3->Text;

else if (RB4->Checked)

return RB4->Text;

return "";

}

bool AllTest::Ch(bool \*mas) {

int z = \*mas;

for (int i = 0; i <= 4; i++)

if (\*(++mas) != z) return false;

return true;

}

void AllTest::P\_Visible(Panel^P1, Panel^P2, Panel^P3) {

P1->Visible = true;

P2->Visible = false;

P3->Visible = false;

}

void AllTest::Panel\_Change(Panel^P1, Panel^P2, Panel^P3, Panel^P7, int \*now\_test) {

switch (now\_test[0])

{

case 0: {

AllTest::P\_Visible(P1, P2, P3);

break; }

case 1: {

AllTest::P\_Visible(P2, P1, P3);

break; }

case 2: {

AllTest::P\_Visible(P3, P2, P1);

break; }

default: {

AllTest::P\_Visible(P7, P2, P1);

P3->Visible = false;

break;

}

}

}

void AllTest::test\_check(bool \*Test\_Types, int \*now\_test, string \*\*PTran, string \*\*vvod, string \*\*Rbutton, Panel ^ p1, Panel ^ p2, Panel^p3, Panel ^ p7, DataGridView ^ dataGridView1, string \*\*all\_ans, int n) {

if (now\_test[0] == 0) {

if (Test\_Types[0] == false)

now\_test[0]++;

else return;

}

if (now\_test[0] == 1) {

if (Test\_Types[1] == false)

now\_test[0]++;

else return;

}

if (now\_test[0] == 2) {

if (Test\_Types[2] == false)

now\_test[0]++;

else return;

}

if (now\_test[0] >= 3) {

MessageBox::Show("Количество правильных ответов: " + Convert::ToString(now\_test[1]), "Правильные ответы", MessageBoxButtons::OK, MessageBoxIcon::Information);

AllTest::P\_Visible(p7, p2, p3);

p1->Visible = false;

AllTest::Output\_arr(all\_ans, n, 5, dataGridView1);

return;

}

}

void AllTest::Output\_arr(string \*\*mas, int n, int m, DataGridView ^ DGV) {

DGV->ColumnHeadersVisible = false;

DGV->RowHeadersVisible = false;

DGV->ColumnCount = m + 1;

DGV->RowCount = n + 1;

DGV->RowHeadersWidth = 10;

DGV->Rows[0]->Cells[0]->Value = "Ответы";

for (int i = 0; i < m; i++)

DGV->Rows[0]->Cells[i + 1]->Value = "[" + (i + 1) + "]";

for (int j = 0; j < n; j++) {

if (j % 2 == 0)

DGV->Rows[j + 1]->Cells[0]->Value = "Пользователя";

else

DGV->Rows[j + 1]->Cells[0]->Value = "Правильный";

}

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++)

DGV->Rows[i + 1]->Cells[j + 1]->Value = ConvertToString(mas[i][j]);

}

int width = 0;

int height = 0;

for (int s = 0; s < DGV->ColumnCount; s++)

width += (DGV->Columns[s]->Width);

if (width > 1300) {

DGV->Width = 1300;

height += 17;

}

else

DGV->Width = width + 3;

for (int z = 0; z < DGV->RowCount; z++)

height += (DGV->Rows[z]->Height);

if (height > 400) {

DGV->Height = 400;

DGV->Width += 17;

}

else

DGV->Height = height + 3;

for (int i = 1; i < n; i+=2) {

for (int j = 1; j < 6; j++) {

if (Convert::ToString(DGV->Rows[i]->Cells[j]->Value) == Convert::ToString(DGV->Rows[i + 1]->Cells[j]->Value)) {

DGV->Rows[i]->Cells[j]->Style->BackColor = System::Drawing::Color::Green;

DGV->Rows[i + 1]->Cells[j]->Style->BackColor = System::Drawing::Color::Green;}

else {

DGV->Rows[i]->Cells[j]->Style->BackColor = System::Drawing::Color::DarkRed;

DGV->Rows[i + 1]->Cells[j]->Style->BackColor = System::Drawing::Color::DarkRed;

}

}

}

}

}

**VAR.cpp**

#include "VAR.h"

bool \*Test\_Types = new bool[10];

int size\_stage;

std::string Login\_Text;

int \*CalendarGlobal = new int[2];

int \*AchivmentCheckGlobal = new int[6];

# **ПРИЛОЖЕНИЕ Б**

Вопросы к тестированию

Задание с переводом фразы

(1) Draft regulations of the Academia.

Перевод: Проект положений академии

(2) The elections of full members.

Перевод: Выборы действительных членов

(3) Provide a stimulating ground.

Перевод: Обеспечить стимулирующую почву

(4) Preliminary work performed.

Перевод: Предварительная работа выполнена

(5) The highest body of the Section.

Перевод: Главный орган секции

Задание с выбором ответов

(1) Пары:

Satellite - Спутник

(2) Пары:

Curiosities - Диковина

(3) Пары:

Chairman - Председатель

(4) Пары:

Headline - Заголовок

(5) Пары:

Committe - Комиссия

Задание с вводом с клавиатуры

(1) A place of study or training in a special field -> Academy или academy

(2) A systematically organized body of knowledge on a particular subject - > Science или science

(3) A person who tekes part in something - > participant или Participant

(4) A meeting where ideas and views on a particular issue can be exchanged - >Forum или forum

(5) The systematic investigation into and study of materials and sources in order to establish facts -> Research или research