Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 05 01 03 «Информационные системы и технологии»

Специализация «издательско-полиграфический комплекс»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОЙ РАБОТЫ**

по дисциплине «Объектно-ориентированное программирование»

Тема «Программное средство “ Клавиатурный тренажер”»

Исполнитель

студент 2 курса группы 2 \_\_\_\_\_\_\_\_\_\_ П.Е.Жаворонок

подпись, дата

Руководитель

доцент, к.т.н. \_\_\_\_\_\_\_\_\_\_\_ Н.В. Пацей

подпись, дата

Курсовая работа защищена с оценкой

Руководитель Н.В. Пацей

подпись

Минск 2015

Оглавление

[**Введение** 3](#_Toc420062735)

[**1.Теоретическое введение** 4](#_Toc420062736)

[**1.1 Объектно-ориентированное программирование** 4](#_Toc420062737)

[**1.2. C# WinForm** 5](#_Toc420062738)

[**1.3. Microsoft SQL Server** 6](#_Toc420062739)

[**1.4. Архитектура платформы .NET Framework** 7](#_Toc420062740)

[**2.** **Постановка задачи и разработка алгоритма решения задачи** 8](#_Toc420062741)

[**3.** **Обзор существующих клавиатурных тренажеров** 10](#_Toc420062742)

[**4.** **Разработка архитектуры** 13](#_Toc420062743)

[**5. Проектирование и создание базы данных** 15](#_Toc420062744)

[**6.** **Руководство пользователя** 16](#_Toc420062745)

[**7. Тестирование и отладка программы** 18](#_Toc420062746)

[**Заключение** 19](#_Toc420062747)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ** 20](#_Toc420062748)

[**Приложение A** 21](#_Toc420062749)

### **Введение**

В современную жизнь все глубже и глубже проникают компьютерные технологии, поэтому приходится осваивать новые навыки при работе на персональном компьютере. Если азы компьютерной грамотности уже присутствуют в личном багаже, наступает время для развития «скорости» и оптимизации работы с персональным компьютером.

И хотя мощность компьютера растет с каждым годом, скорость работы с ним ограничивается лишь человеческим фактором, так как на сегодняшний день еще не придумано более быстрого способа ввода данных в компьютер, чем клавиатурный метод.

По этой причине современный человек вынужден осваивать старинную профессию «машинистки», совершенствуя навыки «слепого десятипальцевого» набора текстов на клавиатуре.

Слепой метод печати - методика набора текста вслепую, используя все пальцы рук. Существует уже более 100 лет. В XIX веке слепым методом печати на печатных машинках обучали машинисток и секретарей. Это позволило сузить использование стенографии, увеличить производительность труда секретарей .

Этот метод требует специального обучения, для чего существуют курсы машинописи, а также бумажные и компьютерные самоучители (последние нередко выполнены в занимательной форме, подобно видеоигре). При слепом методе печати больший упор делается на тактильные ощущения пальцев и мышечную память, чем на зрение.

С годами расположение клавиш «запоминается» пальцами автоматически, и скорость набора текстов у пользователей персонального компьютера (ПК) со стажем становится довольно высокой. Но, как показывает практика, их «слепой метод» оказывается менее эффективным, чем «классический». Чтобы начинающий пользователь (или программист) добавил в свое резюме немаловажную фразу «слепой десятипальцевый набор текстов на клавиатуре», и вообще, существенно ускорил работу на ПК (освобождая время для других занятий), было создано достаточное количество специализированных программ - клавиатурных тренажеров.

## 

## **1.Теоретическое введение**

# **1.1 Объектно-ориентированное программирование**

ООП возникло в результате развития идеологии процедурного программирования, где данные и подпрограммы (процедуры, функции) их обработки формально не связаны. Для дальнейшего развития объектно-ориентированного программирования часто большое значение имеют понятия события (так называемое событийно-ориентированное программирование) и компонента (компонентное программирование, КОП).

Взаимодействие объектов происходит посредством сообщений. Результатом дальнейшего развития ООП, по-видимому, будет агентно-ориентированое программирование, где агенты — независимые части кода на уровне выполнения. Взаимодействие агентов происходит посредством изменения среды, в которой они находятся.

Языковые конструкции, конструктивно не относящиеся непосредственно к объектам, но сопутствующие им для их безопасной (исключительные ситуации, проверки) и эффективной работы, инкапсулируются от них в аспекты (в аспектно-ориентированном программировании). Субъектно-ориентированное программирование расширяет понятие объекта посредством обеспечения более унифицированного и независимого взаимодействия объектов. Может являться переходной стадией между ООП и агентным программированием в части самостоятельного их взаимодействия. Первым языком программирования, в котором были предложены принципы объектной ориентированности, была Симула. В момент его появления в 1967 году в нём были предложены революционные идеи: объекты, классы, виртуальные методы и др., однако это всё не было воспринято современниками как нечто грандиозное. Тем не менее, большинство концепций были развиты Аланом Кэем и Дэном Ингаллсом в языке Smalltalk. Именно он стал первым широко распространённым объектно-ориентированным языком программирования.

В настоящее время количество прикладных языков программирования (список языков), реализующих объектно-ориентированную парадигму, является наибольшим по отношению к другим парадигмам. В области системного программирования до сих пор применяется парадигма процедурного программирования, и общепринятым языком программирования является Cи. При взаимодействии системного и прикладного уровней операционных систем заметное влияние стали оказывать языки объектно-ориентированного программирования. Например, одной из наиболее распространённых библиотек мультиплатформенного программирования является объектно-ориентированная библиотека Qt, написанная на языке.

Сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса или копирования прототипа (например, после запуска результатов компиляции и связывания исходного кода на выполнение).

Прототип — это объект-образец, по образу и подобию которого создаются другие объекты. Объекты-копии могут сохранять связь с родительским объектом, автоматически наследуя изменения в прототипе; эта особенность определяется в рамках конкретного языка.

## **1.2****. C# WinForm**

C# (произносится си-шарп) — язык программирования, сочетающий объектно-ориентированные и контекстно-ориентированные концепции. Разработан в 1998—2001 годах группой инженеров под руководством Андерсa Хейлсбергa в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков C++, Delphi, Modula и Smalltalk — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++) или вывода типов (в отличие от Haskell).

# **1.3.** **Microsoft SQL Server**

SQL Server Management Studio — утилита из [Microsoft SQL Server](http://ru.wikipedia.org/wiki/Microsoft_SQL_Server) 2008 и более поздних версий для конфигурирования, управления и [администрирования](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) всех компонентов Microsoft SQL Server. Утилита включает [скриптовый](http://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) редактор и графическую программу, которая работает с объектами и настройками сервера.

Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью ими управлять.

SQL является прежде всего [информационно-логическим языком](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), предназначенным для описания, изменения и извлечения данных, хранимых в [реляционных базах данных](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). SQL можно назвать [языком программирования](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), при этом он не является [тьюринг-полным](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B0_%D0%BF%D0%BE_%D0%A2%D1%8C%D1%8E%D1%80%D0%B8%D0%BD%D0%B3%D1%83), но вместе с тем стандарт языка спецификацией [SQL/PSM](http://ru.wikipedia.org/wiki/SQL/PSM) предусматривает возможность его [процедурных расширений](http://ru.wikipedia.org/wiki/SQL#.D0.9F.D1.80.D0.BE.D1.86.D0.B5.D0.B4.D1.83.D1.80.D0.BD.D1.8B.D0.B5_.D1.80.D0.B0.D1.81.D1.88.D0.B8.D1.80.D0.B5.D0.BD.D0.B8.D1.8F).

Изначально SQL был основным способом работы пользователя с [базой данных](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и позволял выполнять следующий набор операций:

* создание в базе данных новой таблицы;
* добавление в таблицу новых записей;
* изменение записей;
* удаление записей;
* выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);
* изменение структур таблиц.

При всех своих изменениях SQL остаётся единственным механизмом связи между прикладным программным обеспечением и базой данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов.

Таким образом использование SQL сводится, по сути, к формированию всевозможных выборок строк и совершению операций над всеми записями, входящими в набор.

# **[1.4. Архитектура платформы .NET Framework](javascript:void(0)" \o "Щелкните, чтобы свернуть. Щелкните дважды, чтобы свернуть все.)**

Программа на языке C# выполняется в среде .NET Framework – интегрированном компоненте Windows, содержащем виртуальную систему выполнения (среда CLR) и унифицированный набор библиотек классов. Среда CLR представляет собой коммерческую реализацию Майкрософт инфраструктуры CLI (common language infrastructure), международного стандарта, основы сред выполнения и разработки с тесным взаимодействием языков и библиотек.

Исходный код, написанный на языке C#, компилируется в промежуточный язык (IL) в соответствии со спецификацией CLI. Код IL и ресурсы, такие как растровые изображения и строки, хранятся на диске в исполняемом файле, называемом сборкой, с расширением EXE или DLL в большинстве случаев. Сборка содержит манифест со сведениями о типах сборки, версии, языке и региональных параметрах и требованиях безопасности.

При выполнении программы на C# сборка загружается в среду CLR в зависимости от сведений в манифесте. Далее, если требования безопасности соблюдены, среда CLR выполняет JIT-компиляцию для преобразования кода IL в инструкции машинного кода. Среда CLR также предоставляет другие службы, относящиеся к автоматическому сбору мусора, обработке исключений и управлению ресурсами. Код, выполняемый средой CLR, иногда называют "управляемым кодом" в противопоставление "неуправляемому коду", который компилируется в машинный код, предназначенный для определенной системы. Далее показаны отношения во время компиляции и время выполнения между файлами с исходным кодом C#, библиотеками классов .NET Framework, сборками и средой CLR.

Взаимодействие между языками является ключевой особенностью .NET Framework. Поскольку код на промежуточном языке (IL), создаваемый компилятором C#, соответствует спецификации CTS, код IL на основе C# может взаимодействовать с кодом, создаваемым версиями языков Visual Basic, Visual C++, Visual J# платформы .NET Framework и еще более чем 20 CTS-совместимых языков. В одной сборке может быть несколько модулей, написанных на разных языках платформы .NET Framework, и типы могут ссылаться друг на друга, как если бы они были написаны на одном языке.

Помимо служб времени выполнения, в .NET Framework также имеется обширная библиотека, состоящая из более чем 4000 классов, организованных по пространствам имен, которые обеспечивают разнообразные полезные функции для любых действий, начиная от ввода и вывода файлов для управлением строками для разбивки XML, и заканчивая элементами управления Windows Forms. В обычном приложении на языке C# библиотека классов .NET Framework интенсивно используется для "устройства" кода.

## **Постановка задачи и разработка алгоритма решения задачи**

В рамках курсового проекта необходимо разработать клавиатурный тренажер.

При запуске программа должна автоматически загружать настройки, которые хранятся в соответствующих файлах.

Пользователь должен пройти процедуру аутентификации, указав свой логин. Если пользователь вошел, он может приступить к выполнению заданий, предварительно выбрав одно из предложенных заданий (в системе должны быть предусмотрены 10 заданий различной сложности). Все задания должны храниться в текстовых файлах. При выполнении упражнения должна отображаться виртуальная клавиатура с подсвечивающимся текущим символом. Кроме того должна вестись статистика выполненных упражнений (количество допущенных ошибок по отношению к максимально допустимому количеству ошибок, время выполнения упражнения), которая также должна отображаться при выполнении упражнения.

Начало

Ввод логина

Логин уже использовался

ДА

НЕТ

Начать новое занятие

Начать следующее занятие

Выполнение занятия

Получение результатов

Рисунок 1 – Алгоритм работы с программой

Конец

Выход / Продолжение

## **Обзор существующих клавиатурных тренажеров**

Клавиатурные тренажеры бывают двух видов. Одни выполнены в виде программы, которую можно купить или скачать(она устанавливается на ваш компьютер как любая другая программа). Второй вид тренажеров выполнен в виде on-line сервиса, в котором после регистрации (или без таковой) обучаемому предоставляется возможность научиться печатать слепым методом. Но для того, чтобы им воспользоваться, необходимо подключение Интернет, поэтому для многих все же предпочтительнее обучаться именно на тренажерах-программах.

Приведем несколько самых известных клавиатурных тренажеров.

«Соло на клавиатуре» - самый известный и проверенный временем тренажер работы на клавиатуре компьютера (рисунок 2). Он основан на методике известного психолога, журналиста и преподавателя МГУ Владимира Владимировича Шахиджаняна, собравшего для ее пропаганды и реализации дружный коллектив единомышленников. «Соло на клавиатуре» - это в своем роде уникальный сплав развлекательной, познавательной, развивающей и дисциплинирующей программ. Методика освоения работы с клавиатурой в этой программе, как обычно, следует «от простого к сложному».

Функциональность программы позволяет назвать ее действительно самым совершенным клавиатурным тренажером из ныне существующих. «Соло на клавиатуре» позволяет освоить «скоропись» не только в русской, но и английской раскладке клавиатуры, а также научиться быстро писать транслитерацией. Во время выполнения заданий в окне отображается скорость набора, а по окончании результат заносится в «личное дело».



Рисунок 2 - Тренажер «Соло на клавиатуре»

Известная игровая компания «Дока» внесла свой вклад в массовое освоение клавиатуры - создала игровой клавиатурный тренажер «BabyType 2000» , превращающий урок машинописи в увлекательную игру (рисунок 3).

Идея тренажера - необходимость выполнять упражнения на клавиатуре, чтобы управлять игрой. В процессе игры ставится задача обеспечить безопасное путешествие персонажа по имени Бэбитайп по живописным пространствам, населенным различными опасными существами и механизмами. Бэбитайп может избежать опасности только с помощью обучающегося. Для этого нужно просто убирать буквы, стоящие на его пути, правильно и быстро нажимая соответствующие клавиши на клавиатуре. Программа достаточно эффективно обучает слепому десятипальцевому методу при работе с шестью вариантами раскладки клавиатуры (британской, американской, немецкой, французской, швейцарской и русской).

Клавиатурный тренажер «Stamina» (рисунок 4) обладает вполне внушительным и серьезным набором соответствующих функций, таких как редактор уроков, поддержка множества пользователей, русская и английская, а также альтернативная раскладки клавиатуры, набор текста из внешнего файла, сохранение результатов (скорость и процент ошибок), вывод статистики в виде графика процесса



Рисунок 3 - Тренажер «BabyType 2000»



Рисунок 4 - Тренажер «Stamina»

## **Разработка архитектуры**

Данный курсовой проект основан на диаграмме классов, которая представлена на рисунке ниже:

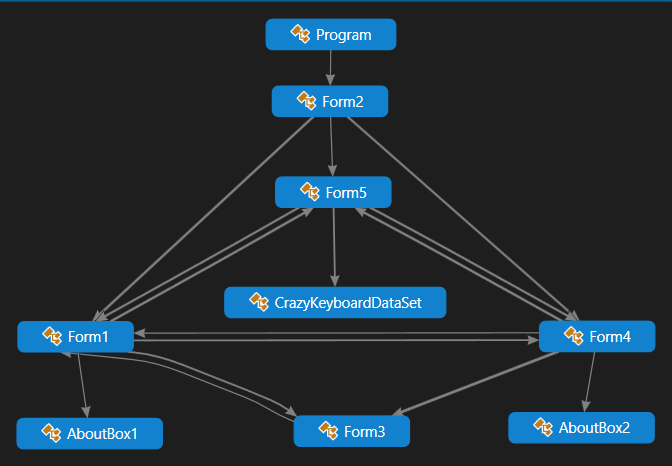


Рисунок 5 – Диаграмма классов

1.Класс Form1

Главный класс с которым и происходит вся работа приложения. В нём находятся все элементы взаимодействия, переходы на другие классы и выполнение заданий пользователем.

2. Класс Form2

Класс предназначен для регистрации и авторизации пользователей. После заполнения формы регистрации данные записываются в базу данных, а при авторизации данные считываются с базы данных и при совпадении логина и происходит переход к главной форме.

3.Класс Form3

Этот класс служит для редактирования занятий пользователем. В нём происходит считывание с txt файлов с занятий и после изменения их, обратно записывать из в txt файлы.

4. Класс Form4

Этот класс является точной копией Form 1 с теми же функциями и возможностями, но на английском языке.

5.Класс Form5

Этот класс предназначен для вывода статистики по всем выполненным заданиям определённым пользователем. В нём пользователь может посмотреть за сколько времени, сколько правильных и неправильных нажатий он совершил во время выполнения занятия.

6. Классы AboutBox1 и AboutBox2

Эти классы предназначены для вывода информации о программе. В них пользователь может посмотреть информацию о номере сборки, названии организации, авторских правах и описании программы.

## **5. Проектирование и создание базы данных**

Выполним физическое проектирование в среде СУБД Microsoft SQL Server. Зададим имена таблиц и полей, определим типы данных и размерность полей таблиц. В таблицах выберем первичные ключи и индексированные поля. Так же для поля определим его основные характеристики – является ли это поле внешним или первичным ключом, создан ли индекс по этому полю, задано ли для поля значение по умолчанию, какие ограничения заданы для данного поля (уникальность значений, маска ввода). Вся эта информация представлена на рисунке 6.

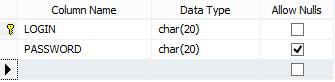


Рисунок 6 – Структура таблиц базы данных «Авторизация»

В таблице представлены следующие столбцы: LOGIN – логин пользователя, PASSWORD – пароль пользователя

А следующая диаграмма иллюстрирует структуру таблицы «Статистика».

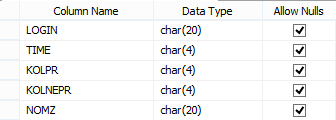


Рисунок 7 – таблица в базе данных

В таблице представлены следующие столбцы: LOGIN – логин пользователя, TIME – время за которое пользователь выполнил упражнение, KOLPR – количество правильных нажатий, KOLNEPR – количество неправильных нажатий, NOMZ – номер выполненного задания.

Взаимосвязь таблиц базы данных представлена на Рисунке 8.

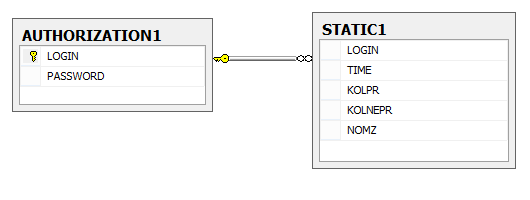


Рисунок 8 – Диаграмма связи таблиц

## **Руководство пользователя**

Для запуска программы необходимо активировать exe – файл с названием «CrazyKeyboard», после этого запустится программа.



Рисунок 9 – Ярлык программы

На заглавной форме присутствует поле для ввода логина и пароля, кнопка, после нажатия которой появляется новое окно.

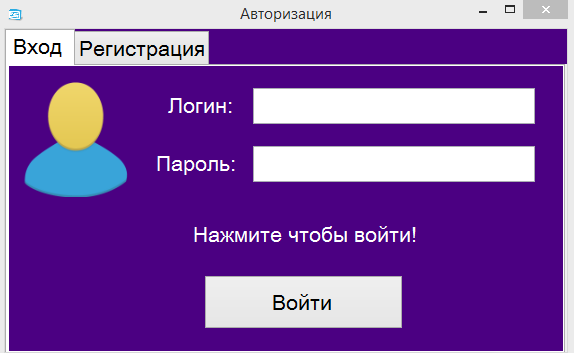


Рисунок 10 – Форма Авторизации

Для того чтобы начать тестирование, необходимо нажать «Войти». Далее появляется главная форма.

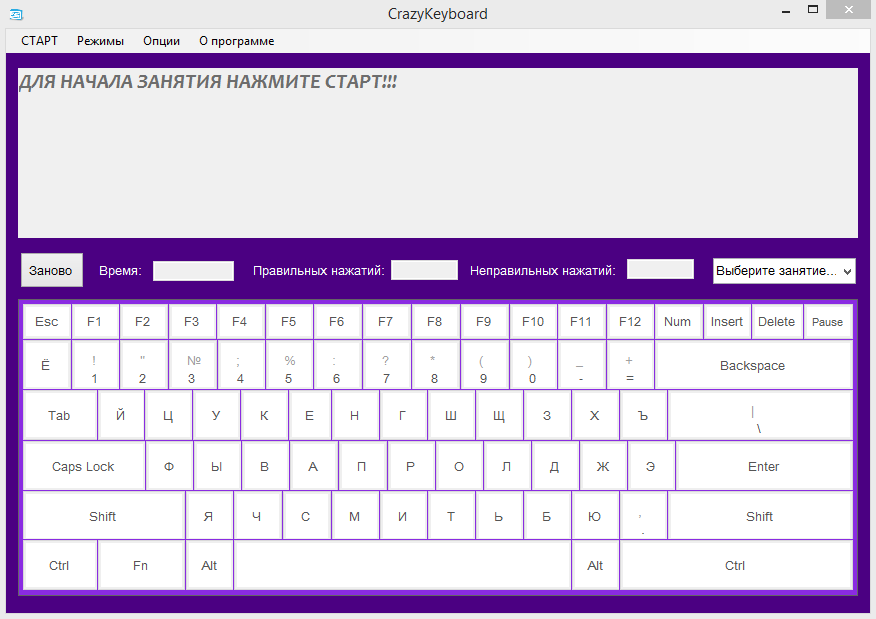


Рисунок 11 – Главная форма

На главной форме есть возможность выбора задания. Можно спрятать клавиатуру или сменить язык формы. Есть возможность редактирования заданий, для этого нужно выбрать: Режимы-> Редактор заданий.

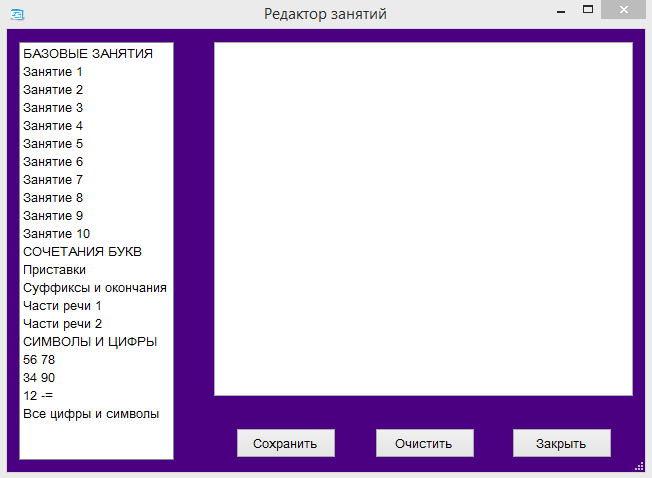


Рисунок 12 – Редактор занятий

После того как будет выбрано занятие для начала выполнения занятия нужно нажать «СТАРТ». После того как будет введён текст, последует окно статистики с результатами.

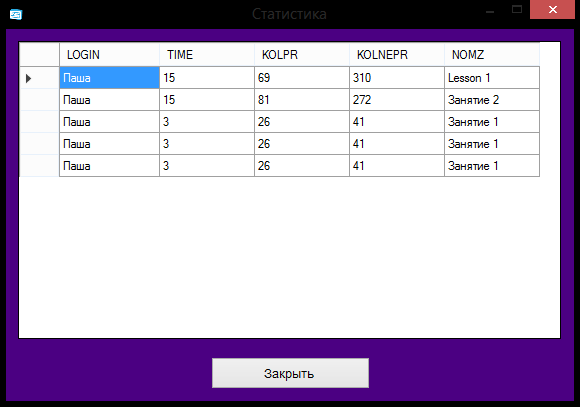


Рисунок 13 – Окно статистики

Пользователю будет предложено выйти из программы, повторить занятие или продолжить выполнение следующих занятий.

# **7.** **Тестирование и отладка программы**

Объектом испытания является программа CrazyKeyboard, находящаяся в папке CrazyKeyboard тренажёр (листинг программы находится в приложении А).

Испытания проводятся с целью проверки правильности функционирования программы и выявления имеющихся недостатков.

Приложение работает в средеWindоws 8.1.

Программу разработана на языке C#.

Для проведения испытания программы Train необходимо загрузить файл исполняемый файл CrazyKeyboard.ехе и проверить работу всех вариантов использования системы. Разработанное приложение может использоваться на персональном компьютере с операционной системой Windows.

Испытания проводятся по следующему алгоритму:

а) запустить на выполнение CrazyKeyboard.ехе, находящийся в папке CrazyKeyboard;

б) в появившейся на экран форме необходимо зарегистрироваться или при имеющейся учётной записи ввести логин и пароль;

в) в появившейся на экран форме необходимо выбрать занятие для выполнения на нажать кнопку «СТАРТ»;

г) после выполнения занятия и просмотра статистики, закрыть окно статистики и посмотреть информацию о программе нажав на кнопку «О программе»;

д) выбрать в меню «Опции» пункт «Раскладка» перейти к английскому языку интерфейса далее выбрать в меню «Mode» пункт «Basic lesson» выбрать занятие и пройти его. Перейти обратно к русской форме.

е) выбрать в меню «Опции» пункт «Раскладка» перейти к английскому языку интерфейса далее выбрать в меню «Mode» пункт «Lesson editor…» выбрать любое задание из списка слева изменить его в правом окне и нажать на кнопку «Save». Перейти обратно к русской форме.

ж) выбрать в меню «Опции» пункт «Раскладка» и убрать галочку с пункта «Клавиатура», пройти задание. После прохождения вернуть клавиатуру обратно.

Проведённые испытания показали, что программа работает верно и без каких-либо ошибок.

## **Заключение**

Решая поставленную задачу, пришли к удовлетворительному результату среди поиска существующих систем. Таким образом, была достигнута цель и был создан клавиатурный тренажёр «CrazyKeyboard».

В программе доступно выполнение различных по уровням сложности упражнений. В ходе выполнения упражнений происходит повышение квалификации обучаемого. Обучаемому доступен просмотр статистики выполненных им упражнений.

Программная система имеет простой, интуитивно понятный интерфейс пользователя.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

## **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Э– Режим доступа: <http://msdn.microsoft.com/library/rus/> -- Дата доступа:14.03.2015.

2. Жарков В Самоучитель Жаркова по анимации и мультипликации в Visual С# .NET Жарков Пресс, 2004. 432 с

3. Либерти Д Программирование на C# Символ-Плюс, 2003. 688 с

4. Майо Дж C#. Искусство программирования ДиаСофт, 2002. 656 с

5. Микелсен К Язык программирования C#. Лекции и упражнения ДиаСофт, 2002. 656 с

6. Петзольд Ч Программирование для Microsoft Windows на С#. В 2 томах М.: Русская редакция, 2002

7. Петзольд Ч Программирование в тональности С#.: Русская редакция, 2004. 512 с

8. Разработка Windows-приложений на Microsoft Visual Basic .NET и Microsoft Visual C# .NET. Учебный курс СПб.: Русская Редакция. 512 с

9. Рихтер Д Программирование на платформе Microsoft .NET Framework СПб.: Русская редакция, 2005. 512 с

10. Робисон У C# без лишних слов ДМК, 2002. 352 с

11. Троелсен Э C# и платформа .NET. Библиотека программиста СПб., 2004. 796 с

12. Шилдт Г C#. Учебный курс СПб., 2002. 511 с

13. Шилдт Г Полный справочник по C# Вильямс, 2004, 752 с

14. Фролов А., Фролов Г Язык C#. Самоучитель : Диалог-МИФИ, 2002. 560 с.

# **Приложение A**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

using System.Media;

using System.Data.SqlClient;

using System.Data.Sql;

namespace CrazyKeyboard

{

public partial class Form1 : Form

{

SqlConnection Connection = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=" + Application.StartupPath + @"\CrazyKeyboard.mdf;Initial Catalog=CrazyKeyboard");

SqlCommand Command = new SqlCommand();

public string login;

private int next = -1;

private Button t;

private int se;

private int correct;

private int incorrect;

private SoundPlayer sp;

private ToolStripMenuItem[] Items;

public Form1()

{

InitializeComponent();

}

public void refresh()

{

timer1.Stop();

txttime.Text = "";

стартToolStripMenuItem.Text = "СТАРТ";

режимыToolStripMenuItem.Enabled = true;

раскладкаToolStripMenuItem.Enabled = true;

comboBox1.Enabled = true;

if (comboBox1.SelectedIndex == 1)

{

reload("lesson1.txt");

}

else if (comboBox1.SelectedIndex == 2)

{

reload("lesson2.txt");

}

else if (comboBox1.SelectedIndex == 3)

{

reload("lesson3.txt");

}

else if (comboBox1.SelectedIndex == 4)

{

reload("lesson4.txt");

}

else if (comboBox1.SelectedIndex == 5)

{

reload("lesson5.txt");

}

else if (comboBox1.SelectedIndex == 6)

{

reload("lesson6.txt");

}

else if (comboBox1.SelectedIndex == 7)

{

reload("lesson7.txt");

}

else if (comboBox1.SelectedIndex == 8)

{

reload("lesson8.txt");

}

else if (comboBox1.SelectedIndex == 9)

{

reload("lesson9.txt");

}

else if (comboBox1.SelectedIndex == 10)

{

reload("lesson10.txt");

}

else if (comboBox1.SelectedIndex == 12)

{

reload("lesson11.txt");

}

else if (comboBox1.SelectedIndex == 13)

{

reload("lesson12.txt");

}

else if (comboBox1.SelectedIndex == 14)

{

reload("lesson13.txt");

}

else if (comboBox1.SelectedIndex == 15)

{

reload("lesson14.txt");

}

else if (comboBox1.SelectedIndex == 17)

{

reload("lesson15.txt");

}

else if (comboBox1.SelectedIndex == 18)

{

reload("lesson16.txt");

}

else if (comboBox1.SelectedIndex == 19)

{

reload("lesson17.txt");

}

else if (comboBox1.SelectedIndex == 20)

{

reload("lesson18.txt");

}

}

public void exit(Form sndr)

{

timer1.Stop();

txttime.Text = "";

txtcorrect.Text = "";

txtincorrect.Text = "";

txtbox.Text = "";

txtbox.Text = "ДЛЯ НАЧАЛА ЗАНЯТИЯ НАЖМИТЕ СТАРТ!!!";

стартToolStripMenuItem.Text = "СТАРТ";

режимыToolStripMenuItem.Enabled = true;

раскладкаToolStripMenuItem.Enabled = true;

comboBox1.Enabled = true;

t.BackColor = Color.White;

}

private void bntf2\_Click(object sender, EventArgs e)

{

}

private void bntesc\_Click(object sender, EventArgs e)

{

}

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void stat()

{

Connection.Close();

Form5 fr5 = new Form5();

fr5.log = login;

SqlCommand Command = new SqlCommand("INSERT INTO STATIC1 (LOGIN, TIME, KOLPR, KOLNEPR, NOMZ) VALUES (@VALUE1, @VALUE2, @VALUE3, @VALUE4, @VALUE5)", Connection);

Command.Parameters.AddWithValue("@VALUE1", login);

Command.Parameters.AddWithValue("@VALUE2", txttime.Text);

Command.Parameters.AddWithValue("@VALUE3", txtcorrect.Text);

Command.Parameters.AddWithValue("@VALUE4", txtincorrect.Text);

if (comboBox1.Text == "Выберите занятие...")

{

if (задание1ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание1ToolStripMenuItem.Text);

}

else if (задание2ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание2ToolStripMenuItem.Text);

}

else if (задание3ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание3ToolStripMenuItem.Text);

}

else if (задание4ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание4ToolStripMenuItem.Text);

}

else if (задание5ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание5ToolStripMenuItem.Text);

}

else if (задание6ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание6ToolStripMenuItem.Text);

}

else if (задание7ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание7ToolStripMenuItem.Text);

}

else if (задание8ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание8ToolStripMenuItem.Text);

}

else if (задание9ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание9ToolStripMenuItem.Text);

}

else if (задание10ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", задание10ToolStripMenuItem.Text);

}

else if (пРИСТАВКИToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", пРИСТАВКИToolStripMenuItem.Text);

}

else if (суффиксыИОкончанияToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", суффиксыИОкончанияToolStripMenuItem.Text);

}

else if (частиРечи1ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", частиРечи1ToolStripMenuItem.Text);

}

else if (частиРечи2ToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", частиРечи2ToolStripMenuItem.Text);

}

else if (toolStripMenuItem2.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", toolStripMenuItem2.Text);

}

else if (toolStripMenuItem3.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", toolStripMenuItem3.Text);

}

else if (toolStripMenuItem4.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", toolStripMenuItem4.Text);

}

else if (всеСимволыИЦифрыToolStripMenuItem.Checked == true)

{

Command.Parameters.AddWithValue("@VALUE5", всеСимволыИЦифрыToolStripMenuItem.Text);

}

else if (comboBox1.Text == "Выберите занятие...")

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 1");

}

}

else

{

if (comboBox1.SelectedIndex == 1)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 1");

}

else if (comboBox1.SelectedIndex == 2)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 2");

}

else if (comboBox1.SelectedIndex == 3)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 3");

}

else if (comboBox1.SelectedIndex == 4)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 4");

}

else if (comboBox1.SelectedIndex == 5)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 5");

}

else if (comboBox1.SelectedIndex == 6)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 6");

}

else if (comboBox1.SelectedIndex == 7)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 7");

}

else if (comboBox1.SelectedIndex == 8)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 8");

}

else if (comboBox1.SelectedIndex == 9)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 9");

}

else if (comboBox1.SelectedIndex == 10)

{

Command.Parameters.AddWithValue("@VALUE5", "Занятие 10");

}

else if (comboBox1.SelectedIndex == 12)

{

Command.Parameters.AddWithValue("@VALUE5", "Приставки");

}

else if (comboBox1.SelectedIndex == 13)

{

Command.Parameters.AddWithValue("@VALUE5", "Суффиксы и окончания");

}

else if (comboBox1.SelectedIndex == 14)

{

Command.Parameters.AddWithValue("@VALUE5", "Части речи 1");

}

else if (comboBox1.SelectedIndex == 15)

{

Command.Parameters.AddWithValue("@VALUE5", "Части речи 2");

}

else if (comboBox1.SelectedIndex == 16)

{

Command.Parameters.AddWithValue("@VALUE5", "56 78");

}

else if (comboBox1.SelectedIndex == 18)

{

Command.Parameters.AddWithValue("@VALUE5", "34 90");

}

else if (comboBox1.SelectedIndex == 19)

{

Command.Parameters.AddWithValue("@VALUE5", "12 -=");

}

else if (comboBox1.SelectedIndex == 20)

{

Command.Parameters.AddWithValue("@VALUE5", "Все цифры и символы");

}

}

Command.Connection.Open();

Command.ExecuteNonQuery();

Command.Connection.Close();

fr5.ShowDialog();

exit(this);

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{ return; }

char key = e.KeyChar;

char[] c = txtbox.Text.ToCharArray();

next++;

//клавиши

for (int i = 0; i < this.panel1.Controls.Count; i++)

{

if (this.panel1.Controls[i].Name.Substring(0, 4).ToLower().CompareTo("bnt" + key.ToString().ToLower()) == 0 && this.panel1.Controls[i].Name.Length <= 4)

highligh((Button)this.panel1.Controls[i], c, next, key);

}

//символы: ',~!, @ #, $ %,...

if ((int)key == 33) highligh(bnt1, c, next, key);

if ((int)key == 64) highligh(bnt2, c, next, key);

if ((int)key == 35) highligh(bnt3, c, next, key);

if ((int)key == 36) highligh(bnt4, c, next, key);

if ((int)key == 37) highligh(bnt5, c, next, key);

if ((int)key == 94) highligh(bnt6, c, next, key);

if ((int)key == 38) highligh(bnt7, c, next, key);

if ((int)key == 42) highligh(bnt8, c, next, key);

if ((int)key == 40) highligh(bnt9, c, next, key);

if ((int)key == 41) highligh(bnt0, c, next, key);

if ((int)key == 126 || (int)key == 96) highligh(bntё, c, next, key);

if ((int)key == 123 || (int)key == 91) highligh(bntх, c, next, key);

if ((int)key == 125 || (int)key == 93) highligh(bntъ, c, next, key);

if ((int)key == 58 || (int)key == 59) highligh(bntж, c, next, key);

if ((int)key == 34 || (int)key == 39) highligh(bntэ, c, next, key);

if ((int)key == 60 || (int)key == 44) highligh(bntб, c, next, key);

if ((int)key == 62 || (int)key == 46) highligh(bntю, c, next, key);

if ((int)key == 63 || (int)key == 47) highligh(bntbackslash, c, next, key);

if ((int)key == 124 || (int)key == 92) highligh(bntforwardslash, c, next, key);

if ((int)key == 95 || (int)key == 45) highligh(bntminus, c, next, key);

if ((int)key == 43 || (int)key == 61) highligh(bntequal, c, next, key);

if ((int)key == 32) highligh(bntspacebar, c, next, key);

if ((int)key == 13) highligh(bntenter, c, next, key);

txtcorrect.Text = correct.ToString();

txtincorrect.Text = incorrect.ToString();

if (c.Length == next + 1)

{

timer1.Stop();

stat();

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

return; }

if (comboBox1.SelectedIndex == 1)

{

reload("lesson1.txt");

}

if (comboBox1.SelectedIndex == 2)

{

reload("lesson2.txt");

}

if (comboBox1.SelectedIndex == 3)

{

reload("lesson3.txt");

}

if (comboBox1.SelectedIndex == 4)

{

reload("lesson4.txt");

}

if (comboBox1.SelectedIndex == 5)

{

reload("lesson5.txt");

}

if (comboBox1.SelectedIndex == 6)

{

reload("lesson6.txt");

}

if (comboBox1.SelectedIndex == 7)

{

reload("lesson7.txt");

}

if (comboBox1.SelectedIndex == 8)

{

reload("lesson8.txt");

}

if (comboBox1.SelectedIndex == 9)

{

reload("lesson9.txt");

}

if (comboBox1.SelectedIndex == 10)

{

reload("lesson10.txt");

}

if (comboBox1.SelectedIndex == 12)

{

reload("lesson11.txt");

}

if (comboBox1.SelectedIndex == 13)

{

reload("lesson12.txt");

}

if (comboBox1.SelectedIndex == 14)

{

reload("lesson13.txt");

}

if (comboBox1.SelectedIndex == 15)

{

reload("lesson14.txt");

}

if (comboBox1.SelectedIndex == 17)

{

reload("lesson15.txt");

}

if (comboBox1.SelectedIndex == 18)

{

reload("lesson16.txt");

}

if (comboBox1.SelectedIndex == 19)

{

reload("lesson17.txt");

}

if (comboBox1.SelectedIndex == 20)

{

reload("lesson18.txt");

}

}

private void Form1\_Load(object sender, EventArgs e)

{

txtbox.BackColor = Color.LightSkyBlue;

textBox1.TabIndex = 0;

se = 0;

correct = 0;

incorrect = 0;

//текст

//таймер

timer1.Interval = 1000;

//звук

sp = new SoundPlayer(@"/BUZZ.WAV");

comboBox1.Items.Add("БАЗОВЫЕ ЗАНЯТИЯ");

comboBox1.Items.Add("Занятие 1");

comboBox1.Items.Add("Занятие 2");

comboBox1.Items.Add("Занятие 3");

comboBox1.Items.Add("Занятие 4");

comboBox1.Items.Add("Занятие 5");

comboBox1.Items.Add("Занятие 6");

comboBox1.Items.Add("Занятие 7");

comboBox1.Items.Add("Занятие 8");

comboBox1.Items.Add("Занятие 9");

comboBox1.Items.Add("Занятие 10");

comboBox1.Items.Add("СОЧЕТАНИЯ БУКВ");

comboBox1.Items.Add("Приставки");

comboBox1.Items.Add("Суффиксы и окончания");

comboBox1.Items.Add("Части речи 1");

comboBox1.Items.Add("Части речи 2");

comboBox1.Items.Add("СИМВОЛЫ И ЦИФРЫ");

comboBox1.Items.Add("56 78");

comboBox1.Items.Add("34 90");

comboBox1.Items.Add("12 -=");

comboBox1.Items.Add("Все цифры и символы");

Items = new ToolStripMenuItem[18];

Items[0] = задание1ToolStripMenuItem;

Items[1] = задание2ToolStripMenuItem;

Items[2] = задание3ToolStripMenuItem;

Items[3] = задание4ToolStripMenuItem;

Items[4] = задание5ToolStripMenuItem;

Items[5] = задание6ToolStripMenuItem;

Items[6] = задание7ToolStripMenuItem;

Items[7] = задание8ToolStripMenuItem;

Items[8] = задание9ToolStripMenuItem;

Items[9] = задание10ToolStripMenuItem;

Items[10] = приставкиToolStripMenuItem1;

Items[11] = суффиксыИОкончанияToolStripMenuItem;

Items[12] = частиРечи1ToolStripMenuItem;

Items[13] = частиРечи2ToolStripMenuItem;

Items[14] = toolStripMenuItem2;

Items[15] = toolStripMenuItem3;

Items[16] = toolStripMenuItem4;

Items[17] = всеСимволыИЦифрыToolStripMenuItem;

}

private void highligh(Button con, char[] c, int next, char key)

{

try

{

t.BackColor = Color.White;

}

catch (Exception e) { }

try

{

if ((int)key == (int)c[next])

{

con.BackColor = Color.Green;

t = con;

txtbox.SelectionStart = next;

txtbox.SelectionLength = 1;

txtbox.SelectionColor = Color.Green;

correct++;

}

else if ((int)key == 13 && (int)c[next] == 10)

{

bntenter.BackColor = Color.Green;

t = bntenter;

correct++;

}

else

{

// ошибка нажатия

con.BackColor = Color.Red;

t = con;

txtbox.SelectionStart = next;

txtbox.SelectionLength = 1;

txtbox.SelectionColor = Color.Red;

incorrect++;

SystemSounds.Exclamation.Play();

}

}

catch (Exception e) { }

}

private void button3\_Click(object sender, EventArgs e)

{

refresh();

timer1.Stop();

}

private void label22\_Click(object sender, EventArgs e)

{

}

private void timer1\_Tick(object sender, EventArgs e)

{

se += 1;

txttime.Text = se.ToString();

}

private string loadtext(string filename)

{

FileStream fs;

StreamReader fr;

fs = new FileStream(filename, FileMode.Open, FileAccess.Read);

fr = new StreamReader(fs);

string content = fr.ReadToEnd();

fs.Close();

return content;

}

private void reload(string filename)

{

try

{

t.BackColor = Color.White;

}

catch (Exception ex) { }

next = -1;

se = 0;

correct = 0;

incorrect = 0;

txtincorrect.Text = "";

txtcorrect.Text = "";

timer1.Start();

textBox1.Focus();

txtbox.Clear();

txtbox.Text = loadtext(filename);

}

private void оПрограммеToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТОП")

{

режимыToolStripMenuItem.Enabled = false;

раскладкаToolStripMenuItem.Enabled = false;

}

else

{

режимыToolStripMenuItem.Enabled = true;

раскладкаToolStripMenuItem.Enabled = true;}

AboutBox1 ab1 = new AboutBox1();

ab1.ShowDialog();

}

private void задание1ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(0);

return;

}

reload("lesson1.txt");

}

private void задание2ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(1);

return;

}

reload("lesson2.txt");

}

private void задание3ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(2);

return;

}

reload("lesson3.txt");

}

private void задание4ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(3);

return;

}

reload("lesson4.txt");

}

private void задание5ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(4);

return; }

reload("lesson5.txt");

}

private void задание6ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(5);

return; }

reload("lesson6.txt");

}

private void задание7ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(6);

return; }

reload("lesson7.txt");

}

private void задание8ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(7);

return; }

reload("lesson8.txt");

}

private void задание9ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(8);

return; }

reload("lesson9.txt");

}

private void задание10ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(9);

return; }

reload("lesson10.txt");

}

private void клавиатураToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (клавиатураToolStripMenuItem.Checked)

{

Изменение\_размера\_окна\_ВНИЗ(880, 308, 623);

}

else

{ Изменение\_размера\_окна\_ВВЕРХ(880, 623, 308); }

}

private void редакторЗаданийToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form3 fr3 = new Form3("r");

string[] myList = new string[21];

myList[0] = "БАЗОВЫЕ ЗАНЯТИЯ";

myList[1] = "Занятие 1";

myList[2] = "Занятие 2";

myList[3] = "Занятие 3";

myList[4] = "Занятие 4";

myList[5] = "Занятие 5";

myList[6] = "Занятие 6";

myList[7] = "Занятие 7";

myList[8] = "Занятие 8";

myList[9] = "Занятие 9";

myList[10] = "Занятие 10";

myList[11] = "СОЧЕТАНИЯ БУКВ";

myList[12] = "Приставки";

myList[13] = "Суффиксы и окончания";

myList[14] = "Части речи 1";

myList[15] = "Части речи 2";

myList[16] = "СИМВОЛЫ И ЦИФРЫ";

myList[17] = "56 78";

myList[18] = "34 90";

myList[19] = "12 -=";

myList[20] = "Все цифры и символы";

fr3.listBox1.Items.AddRange(myList);

fr3.Owner = this;

fr3.ShowDialog();

}

private void русскаяToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void английскаяToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form5 fr5 = new Form5();

Form4 fr4 = new Form4();

fr4.login2 = login;

this.Hide();

fr5.flag = 4;

fr4.ShowDialog();

this.Close();

}

private System.Threading.Thread thread;

public void Изменение\_размера\_окна\_ВНИЗ(int X, int Y, int YChange)

{

Point point = new Point(X, Y);

this.Size = new Size(point);

thread = new System.Threading.Thread(new System.Threading.ThreadStart(delegate()

{

while (point.Y <= YChange)

{

this.Invoke(new Action(delegate()

{

point.Y +=5;

this.Size = new Size(point);

}));

System.Threading.Thread.Sleep(1);

}

}));

thread.Start();

}

public void Изменение\_размера\_окна\_ВВЕРХ(int X, int Y, int YChange)

{

Point point = new Point(X, Y);

this.Size = new Size(point);

thread = new System.Threading.Thread(new System.Threading.ThreadStart(delegate()

{

while (point.Y >= YChange)

{

this.Invoke(new Action(delegate()

{

point.Y-=5;

this.Size = new Size(point);

}));

System.Threading.Thread.Sleep(1);

}

}));

thread.Start();

}

private void Unchecked(int index)

{

for (int i = 0; i < Items.Length; i++)

Items[i].Checked = false;

if (index != -1)

Items[index].Checked = true;

comboBox1.SelectedIndex = -1;

comboBox1.Text = "Выберите занятие...";

}

private void стартToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

режимыToolStripMenuItem.Enabled = false;

раскладкаToolStripMenuItem.Enabled = false;

comboBox1.Enabled = false;

if (comboBox1.SelectedIndex == 1)

{

reload("lesson1.txt");

}

else if (comboBox1.SelectedIndex == 2)

{

reload("lesson2.txt");

}

else if (comboBox1.SelectedIndex == 3)

{

reload("lesson3.txt");

}

else if (comboBox1.SelectedIndex == 4)

{

reload("lesson4.txt");

}

else if (comboBox1.SelectedIndex == 5)

{

reload("lesson5.txt");

}

else if (comboBox1.SelectedIndex == 6)

{

reload("lesson6.txt");

}

else if (comboBox1.SelectedIndex == 7)

{

reload("lesson7.txt");

}

else if (comboBox1.SelectedIndex == 8)

{

reload("lesson8.txt");

}

else if (comboBox1.SelectedIndex == 9)

{

reload("lesson9.txt");

}

else if (comboBox1.SelectedIndex == 10)

{

reload("lesson10.txt");

}

else if (comboBox1.SelectedIndex == 12)

{

reload("lesson11.txt");

}

else if (comboBox1.SelectedIndex == 13)

{

reload("lesson12.txt");

}

else if (comboBox1.SelectedIndex == 14)

{

reload("lesson13.txt");

}

else if (comboBox1.SelectedIndex == 15)

{

reload("lesson14.txt");

}

else if (comboBox1.SelectedIndex == 17)

{

reload("lesson15.txt");

}

else if (comboBox1.SelectedIndex == 18)

{

reload("lesson16.txt");

}

else if (comboBox1.SelectedIndex == 19)

{

reload("lesson17.txt");

}

else if (comboBox1.SelectedIndex == 20)

{

reload("lesson18.txt");

}

else if (comboBox1.Text == "Выберите занятие...")

{

if (задание1ToolStripMenuItem.Checked == true)

{

reload("lesson1.txt");

}

else if (задание2ToolStripMenuItem.Checked == true)

{

reload("lesson2.txt");

}

else if (задание3ToolStripMenuItem.Checked == true)

{

reload("lesson3.txt");

}

else if (задание4ToolStripMenuItem.Checked == true)

{

reload("lesson4.txt");

}

else if (задание5ToolStripMenuItem.Checked == true)

{

reload("lesson5.txt");

}

else if (задание6ToolStripMenuItem.Checked == true)

{

reload("lesson6.txt");

}

else if (задание7ToolStripMenuItem.Checked == true)

{

reload("lesson7.txt");

}

else if (задание8ToolStripMenuItem.Checked == true)

{

reload("lesson8.txt");

}

else if (задание9ToolStripMenuItem.Checked == true)

{

reload("lesson9.txt");

}

else if (задание10ToolStripMenuItem.Checked == true)

{

reload("lesson10.txt");

}

else if (пРИСТАВКИToolStripMenuItem.Checked == true)

{

reload("lesson11.txt");

}

else if (суффиксыИОкончанияToolStripMenuItem.Checked == true)

{

reload("lesson12.txt");

}

else if (частиРечи1ToolStripMenuItem.Checked == true)

{

reload("lesson13.txt");

}

else if (частиРечи2ToolStripMenuItem.Checked == true)

{

reload("lesson14.txt");

}

else if (toolStripMenuItem2.Checked == true)

{

reload("lesson15.txt");

}

else if (toolStripMenuItem3.Checked == true)

{

reload("lesson16.txt");

}

else if (toolStripMenuItem4.Checked == true)

{

reload("lesson17.txt");

}

else if (всеСимволыИЦифрыToolStripMenuItem.Checked == true)

{

reload("lesson18.txt");

}

else

reload("lesson1.txt");

}

timer1.Start();

стартToolStripMenuItem.Text = "СТОП";

}

else

{

MessageBox.Show("Время: " + txttime.Text + " сек." + "\r\n" + "Количество правильных нажатий: " + txtcorrect.Text + "\r\n" + "Количестов неправильных нажатий: " + txtincorrect.Text);

стартToolStripMenuItem.Text = "СТОП";

раскладкаToolStripMenuItem.Enabled = true;

режимыToolStripMenuItem.Enabled = true;

timer1.Stop();

comboBox1.Enabled = true;

стартToolStripMenuItem.Text = "СТАРТ";

}

}

private void приставкиToolStripMenuItem1\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(10);

return; }

reload("lesson11.txt");

}

private void суффиксыИОкончанияToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(11);

return; }

reload("lesson12.txt");

}

private void частиРечи1ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(12);

return; }

reload("lesson13.txt");

}

private void частиРечи2ToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(13);

return; }

reload("lesson14.txt");

}

private void базоToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void режимыToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void раскладкаToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void toolStripMenuItem2\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{ Unchecked(14);

return; }

reload("lesson15.txt");

}

private void toolStripMenuItem3\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(15);

return; }

reload("lesson16.txt");

}

private void toolStripMenuItem4\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(16);

return; }

reload("lesson17.txt");

}

private void всеСимволыИЦифрыToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (стартToolStripMenuItem.Text == "СТАРТ")

{

Unchecked(17);

return; }

reload("lesson18.txt");

}

}

}