**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГБОУ ВО «МАГНИТОГОРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Г.И. НОСОВА»**

Кафедра информатики и информационной безопасности

**КУРСОВАЯ РАБОТА**

по дисциплине «Языки программирования»

на тему: «Разработка пользовательского Windows-приложения на языке C#»

Исполнитель: Барашков С.А. студент 2 курса, группа АИБ-19-1

Руководитель: Михайлова У.В. доц. каф. ИиИБ, к.т.н.

Работа допущена к защите « »января 2021 г.

(подпись)

Работа защищена « »января 2021 г. с оценкой

(подпись)

(оценка)

Магнитогорск, 2021

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГБОУ ВО «МАГНИТОГОРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Г.И. НОСОВА»**

Кафедра информатики и информационной безопасности

**ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Тема: «Разработка пользовательского Windows-приложения на языке C#»

Студенту Барашкову Сергею Александровичу

Исходные данные:

Индивидуальное задание на курсовую работу; требования к входным и выходным данным; лекционные и справочные /материалы; среда программирования MS Visual Studio 2019; интернет ресурсы, UML диаграммы классов.

Срок сдачи: « »января 2021 г.

Руководитель: / Михайлова У.В.

Задание получил: / Барашков С.А.

Магнитогорск, 2021

Содержание

[Введение 4](#_Toc62580579)

[1.Краткий разбор среды разработки 5](#_Toc62580580)

[1.1.Общие сведения о языке C# 5](#_Toc62580581)

[1.2.Классы и объекты 9](#_Toc62580582)

[1.3. Сфера применения C# 11](#_Toc62580583)

[1.4.Среда разработки 12](#_Toc62580584)

[1.4.1 Возможности среды разработки 13](#_Toc62580585)

[1.4.2 Создание программ в Visualstudio 2019 14](#_Toc62580586)

[2.Разработка приложения 17](#_Toc62580587)

[2.1.Постановка задачи 17](#_Toc62580588)

[2.2 Обзор реализации проекта 19](#_Toc62580589)

[2.2.1 Компоненты формы 19](#_Toc62580590)

[2.2.2 Классы 27](#_Toc62580591)

[2.3.Руководство пользователя 33](#_Toc62580592)

[Заключение 37](#_Toc62580593)

[Список литературы 38](#_Toc62580594)

[Приложение А 39](#_Toc62580595)

[Приложение Б 42](#_Toc62580596)

[Приложение В 43](#_Toc62580597)

[Приложение Г 57](#_Toc62580598)

[Приложение Д 60](#_Toc62580599)

[Приложение Е 63](#_Toc62580600)

# Введение

Предлагаемая исследовательская работа посвящена созданию приложений на языке программирования С# в интегрированной среде разработки Visual Studio. На момент 2021 года данный язык программирования позволяет реализовать любое приложение, игру или программу. Данная работа призвана показать возможности С# при создании индивидуальных программ.

Данная программа реализована в виде опроса, в котором пользователь самостоятельно вводит данные для дальнейшей работы программы. Следует заметить, что пользователь может вводить совершенно неподходящие для выполнения задания данные, которые программа должна обрабатывать правильно.

Для решения поставленной задачи (реализации программы) необходимо выполнить следующие подзадачи:

1. Спроектировать иерархию классов;

2. Реализовать правильные переходы между существующими формами;

3. Придумать и реализовать нужные вопросы с доступными ответами;

4. Найти способ эффективно по времени реализовать сохранение всех ответов пользователя;

5. Предоставить пользователю понятную информацию по завершении тестирования.

# 1.Краткий разбор среды разработки

## 1.1.Общие сведения о языке C#

C# (си шарп) — объектно-ориентированный язык программирования, разработанный в 1998—2001 годах группой инженерами компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота. C# - язык разработки приложений для платформы Microsoft .NET Framework.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

C# — это объектно- и компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО.

Вот лишь несколько функций языка C#, обеспечивающих надежность и устойчивость приложений:

* Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами.
* Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них.
* Лямбда-выражения поддерживают приемы функционального программирования.
* Синтаксис запросов создает общий шаблон для работы с данными из любого источника.
* Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем.
* Сопоставление шаблонов предоставляет синтаксис для простого разделения данных из алгоритмов в современных распределенных системах.

В C# действует единая система типов. Все типы C#, включая типы-примитивы, такие как «int» и «double», наследуют от одного корневого типа «object». Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. Он также позволяет динамически выделять объекты и хранить упрощенные структуры в стеке.

В C# особое внимание уделяется управлению версиями для обеспечения совместимости программ и библиотек при их изменении. Вопросы управления версиями существенно повлияли на такие аспекты разработки C#, как раздельные модификаторы «virtual» и «override», правила разрешения перегрузки методов и поддержка явного объявления членов интерфейса. Версии языка приведены ниже в таблице 1.

Таблица 1 – Версии языка C#

|  |  |
| --- | --- |
| **Версия** | **Нововведения** |
| **C# 2.0** | * Частичные типы * Обобщённые типы «*generics*» * Итераторы и ключевое слово »yield» * Анонимные методы * Оператор »*null»*-объединения * *«Nullable»*-типы |
| **C# 3.0** | * Запросы, интегрированные в язык (*LINQ*) * Инициализаторы объектов и коллекций * Лямбда-выражения * Деревья выражений * Неявная типизация и ключевое слово »var» * Анонимные типы * Методы расширения * Автоматические свойства |
| **C# 4.0** | * Динамическое связывание и ключевое слово »dynamic» * Именованные и опциональные аргументы * Обобщенная [ковариантность и контрвариантность](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B8_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%B0%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_(%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%D0%B5)) * Библиотека *TPL*, концепция задач и классы Task, Parallel * Класс MemoryCache * Классы параллельных коллекций |
| **C# 5.0** | * Шаблон *TAP* * Асинхронные методы »async» и »await» * Сведения о вызывающем объекте |
| **C# 6.0** | * Компилятор как сервис * Импорт членов статических типов в пространство имён * Фильтры исключений * «await» вблоках catch/finally * Инициализаторы автосвойств * Автосвойства только для чтения * *«null»*-условные операции (?. и ?[]) * Интерполяция строк * Оператор nameof * Инициализатор словаря * Функции сжатые до выражений |
| **C# 7.0**[]](https://ru.wikipedia.org/wiki/C_Sharp#cite_note-seven-17) | * out-переменные * Сопоставление с шаблоном * Шаблоны с is * Шаблоны и выражение switch * Кортежи * Распаковка кортежей (деконструкторы) * Локальные функции * Улучшения литералов * Локальные переменные и возвращаемые значения по ссылке * Расширение списка типов, возвращаемых асинхронными методами * Больше членов класса в виде выражений * throw выражения |
| **C# 8.0** | * Члены только для чтения * Члены интерфейса по умолчанию * Улучшения сопоставления шаблонов * Объявления using * Статические локальные функции * Удаляемые ссылочные структуры * Ссылочные типы, допускающие значение NULL * Асинхронные потоки * Индексы и диапазоны * Присваивание объединения со значением NULL * Неуправляемые сконструированные типы * Выражения stackalloc во вложенных выражениях * Больше членов класса в виде выражений * Улучшения интерполированных строк |

В таблице выше отражены все нововведения по каждой из версий языка, с кратким описанием каждого пункта. Таблица показывает как развивался и расширялся спектр возможностей языка с течением времени.

## 1.2.Классы и объекты

Класс — это конструктор, который позволяет создавать свои собственные пользовательские типы путем группирования переменных других типов, методов и событий. Класс подобен чертежу. Он определяет данные и поведение типа. Если класс не объявлен статическим, то клиентский код может его использовать, создав объекты или экземпляры, назначенные переменной. Переменная остается в памяти, пока все ссылки на нее не выйдут из области видимости. В это время среда CLR помечает ее пригодной для сборщика мусора. Если класс объявляется статическим, то в памяти остается только одна копия и клиентский код может получить к ней доступ только посредством самого класса, а не переменной экземпляра.

В отличие от структур классы поддерживают наследование, фундаментальную характеристику объектно-ориентированного программирования.

Объявление класса

Классы объявляются  с помощью ключевого слова class  
*Пример:* public class Customer

     {

         //Fields, properties, methods and events go here...

     }

Ключевому слову class предшествует уровень доступа. Поскольку в данном случае используется public, любой может создавать объекты из этого  класса. По умолчанию классы определяются как internal (внутренние), что означает, что доступ к ним будет иметь только код текущего проекта. При желании можно указать, что класс является общим и должен быть доступен из других проектов. Для этого используется ключевое слово public. Классы, которые объявляются самостоятельно, не могут быть частными или защищенными. Соответствующие модификаторы — private и protected — можно использовать только для описания классов, являющихся членами других классов.

Имя класса указывается после ключевого слова class. Оставшаяся часть определения является телом класса, в котором задаются данные и поведение. Поля, свойства, методы и события в классе обозначаются термином члены класса.

Класс и объект — это разные вещи, хотя в некоторых случаях они взаимозаменяемы. Класс определяет тип объекта, но не сам объект. Объект — это конкретная сущность, основанная на классе и иногда называемая экземпляром класса.

Объекты можно создавать с помощью ключевого слова new, за которым следует имя класса, на котором будет основан объект:

Пример: Customer object1 = new Customer();

При создании экземпляра класса ссылка на этот объект передается программисту. В предыдущем примере object1 является ссылкой на объект, основанный на Customer. Эта ссылка указывает на новый объект, но не содержит данные этого объекта. Фактически, можно создать ссылку на объект без создания самого объекта:

Пример: Customer object2;

Создание таких ссылок, которые не указывают на объект, не рекомендуется, так как попытка доступа к объекту по такой ссылке приведет к сбою во время выполнения. Однако такую ссылку можно сделать указывающей на объект, создав новый объект или назначив ее существующему объекту:

Пример: Customer object3 = new Customer();

Customer object4 = object3;

В данном коде создаются две ссылки на объекты, которые указывают на один объект. Поэтому любые изменения объекта, выполненные посредством object3, будут видны при последующем использовании object4.Поскольку на объекты, основанные на классах, указывают ссылки, классы называют ссылочными типами.

## 1.3. Сфера применения C#

С# — один из языков программирования, который может использоваться для создания приложений. Создавать приложения на С# легче, чем на C+ +, поскольку синтаксис языка С# более простой, чем синтаксис C++.

Некоторые наиболее часто встречающиеся типы приложений:

Приложения Windows. Это приложения вроде Microsoft Office, имеющие знакомый "Windows-подобный" вид и представление. Создавать такие приложения достаточно просто с помощью модуля .NET Framework, который называется Windows Forms и представляет собой библиотеку управляющих элементов (кнопок, панелей инструментов, меню и т. п.); эта библиотека может использоваться для создания пользовательского интерфейса (user interface, UI) Windows.

Web-приложения. Эти приложения представляют собой web-страницы, которые могут просматриваться любым web-браузером. В состав .NET Framework входит мощная система динамического создания содержимого web-страниц, позволяющая идентифицировать пользователя, обеспечивать безопасность и пр. Эта система называется Active Server Pages.NET (ASP.NET — активные серверные страницы .NET); для создания приложений ASP.NET можно применять Web Forms языка С#(Шилдт, 2009).

Web-службы. Это новый замечательный способ создания гибких распределенных приложений. С помощью web-служб можно обмениваться по Интернету практически любыми данными с использованием единого простого синтаксиса независимо от того, какой язык программирования применялся при создании web-службы и на какой системы она размещена. Приложениям всех перечисленных типов может потребоваться доступ к базам данных, что осуществляется с помощью раздела .NET Framework, называемого Active Data Objects.NET (ADO.NET — активные объекты с данными .NET). Также можно использовать и многие другие ресурсы, например, инструменты для создания сетевых компонентов, графического вывода, выполнения сложных математических вычислений и т.д.

## 1.4.Среда разработки

[1]Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

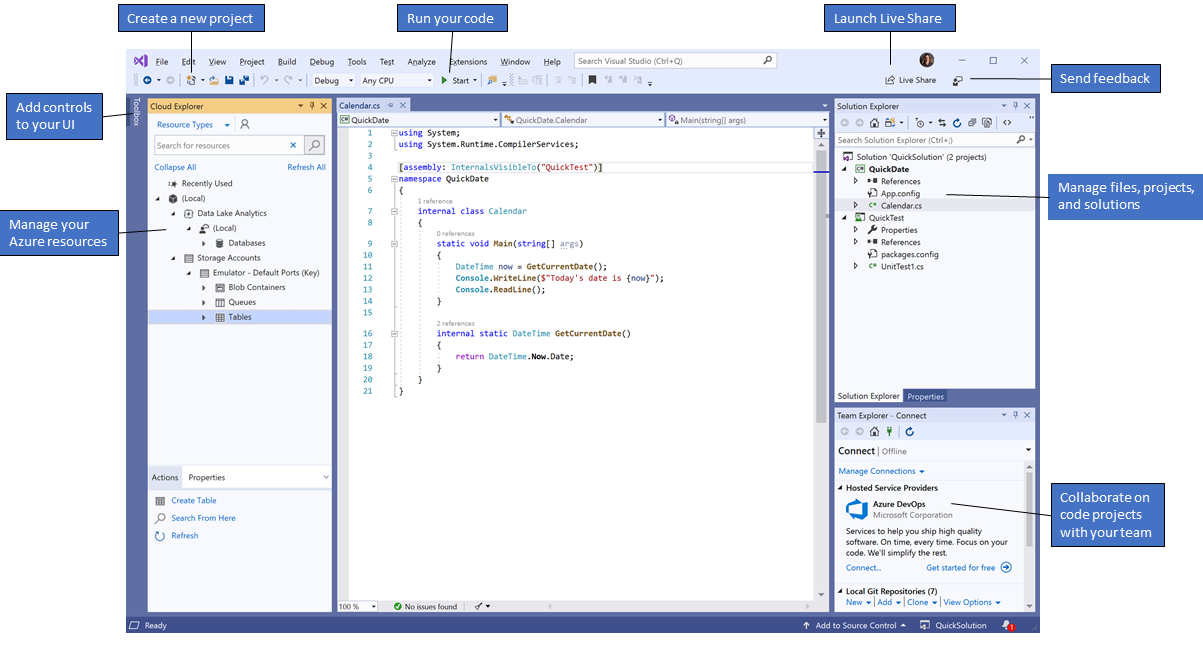


Рисунок 1.1 – Внешний вид VisualStudio 2019

На рисунке показана среда Visual Studio с открытым проектом и несколькими окнами основных инструментов, которые наиболее часто используются в разработке:

* Обозреватель решений (вверху справа) позволяет просматривать файлы кода, перемещаться по ним и управлять ими. Обозреватель решений позволяет упорядочить код путем объединения файлов в решения и проекты.
* В окне редактора (центр), где вы, скорее всего, будете проводить большую часть времени, отображается содержимое файла. Здесь вы можете редактировать код или разрабатывать пользовательский интерфейс, например окно с кнопками или текстовые поля.
* Team Explorer (правый нижний угол) позволяет отслеживать рабочие элементы и использовать код совместно с другими пользователями с помощью технологий управления версиями, таких как Git и система управления версиями Team Foundation (TFVC).

### 1.4.1 Возможности среды разработки

* [2] IntelliSense. Технология автодополнения Microsoft. Дописывает название функции при вводе начальных букв. Кроме прямого назначения, IntelliSense используется для доступа к документации и для устранения неоднозначности в именах переменных, функций и методов, используя рефлексию.
* Code Anilizer. Функционал, который помогает найти ошибки в коде. Совмещён с IntelliSense, тем, что все ошибки, уведомления, потенциальные ошибки подсвечиваются. Также вся эта информация отображается в окне «Error List».
* Perfomance Analizer. Инструмент, отображающий затраты ресурсов при работе приложения/сервиса в виде статистики и графиков.
* [3]Test Manager. Встроенный менеджер тестов. После создания теста можно с помощью специального окна запускать и настраивать тесты.
* Extension/Updates Manager. Менеджер плагинов, адаптеров, провайдеров. Позволяет легко найти, установить, обновить любое дополнение.
* Nuget. Система управления пакетами для платформ разработки Microsoft, в первую очередь библиотек .NET Framework. Управляется .NET Foundation. Удобная установка библиотек в любой .Net проект.
* Git Manager. Встроенный менеджер контроля версий. Изначально работал только с Team Foundation Server. Сейчас можно подключить Team Explorer (Название менеджера) к любому репозиторию. Присутствуют все необходимые функции для работы с git без запросов.
* Archivator. Архиватор проектов. После того, как проект готов, нужно собрать исполняемый файл. Для каждой технологии реализован свой архиватор. Не нужно устанавливать отдельный софт, чтобы сделать установочник.
* File Manager. Для добавления нового файла в проект существует встроенный менеджер файлов. Удобное создание любых файлов на основе шаблонов. Реализовано большое количество стандартных шаблонов (Пример: класс). Также можно добавлять свои. При установке новой технологии - добавляются соответствующие шаблоны.
* Views. Большое количество различных вкладок для отображения различной полезной информации, вроде списка «GOTO», или отображения данных объекта в Debug режиме.
* Customization. Возможность изменить внешний вид Visual Studio под себя. Изменения цветов, темы, шрифтов, отступов и т.д. Расположение окон в удобном вам виде.
* Setting. Настройка всего выше перечисленного функционала. Настройка быстрых клавиш, уведомлений, быстрый запуск, стартового окна, вкладок, разметки языков и много другого
* Благодаря огромному количеству настроек, поддерживаемых технологий, быстродействию и удобству Visual Studio считается одной из лучших сред разработки. Из минусов можно выделить огромный вес пакетов технологий.

### 1.4.2 Создание программ в Visualstudio 2019

[4] Создание программы начинается с запуска программы и выбора необходимой вкладки на стартовом экране (Рис 1.2).

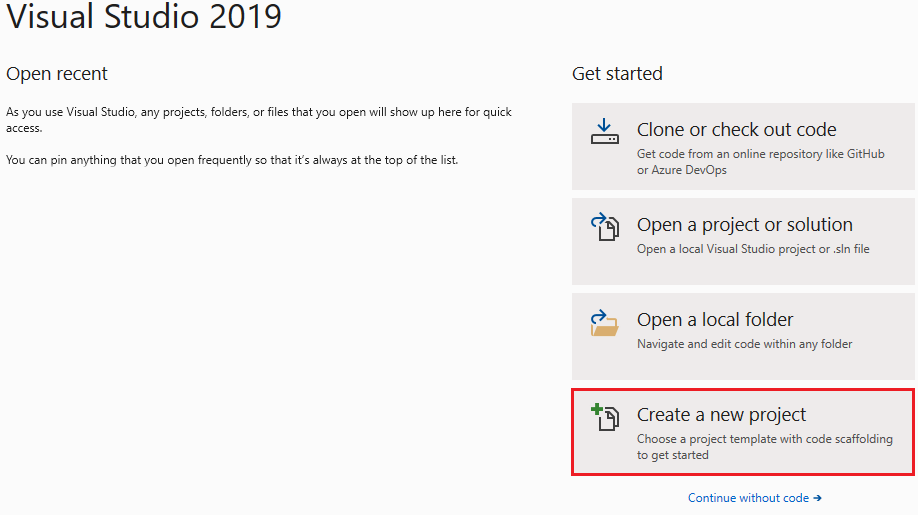


Рисунок 1.2 – Начальное окно

При запуске отображается начальное окно, где можно клонировать репозиторий, открыть недавно использованные проекты или создать проект.

После нажатия откроется окно создания проекта с отображением нескольких шаблонов проектов (Рис 1.3). Шаблон содержит основные файлы и параметры, необходимые для данного типа проекта.Чтобы найти нужный шаблон, необходимо ввести консоль .net core в поле поиска. Список доступных шаблонов автоматически отфильтруется по введенным словам. Также можно выбрать наиболее часто использующийся шаблон.

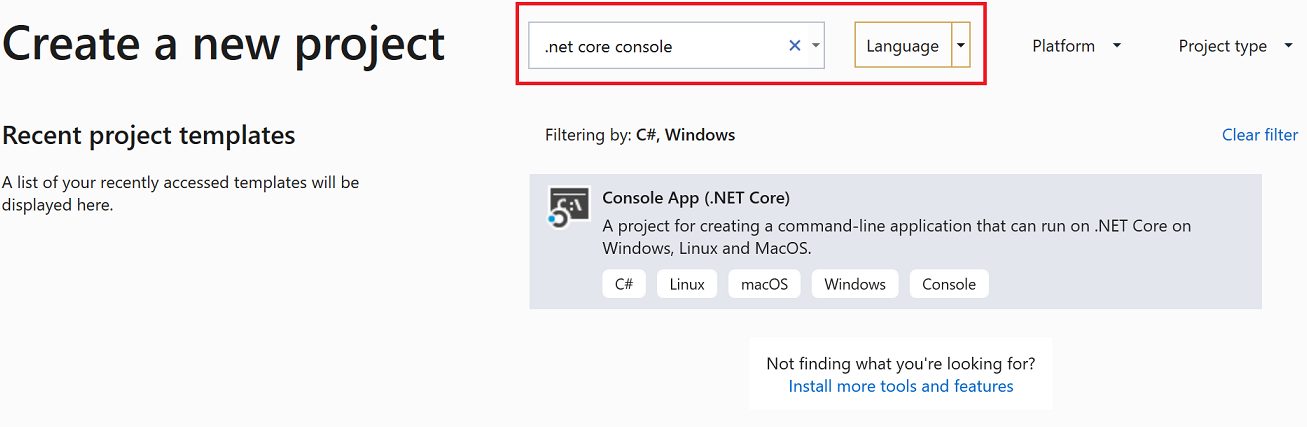


Рисунок 1.3 – Окно создания проекта

Далее пользователь вводит название проекта и его расположение в системе. Расположение можно оставить по умолчанию, тогда проект будет сохранен в папку со средой разработки VisualStudio. В противном случае, если пользователю необходимо сохранить проект на сторонний носитель, другую папку и т.д, ему будет достаточно лишь указать необходимое расположение.

Когда все необходимые поля заполнены, по заранее выбранному шаблону создастся пустой проект и откроется окно работы с проектом

Код C# для приложения отображается в окне редактора, который занимает большую часть пространства. Обратите внимание, что текст автоматически выделяется цветом для обозначения разных частей кода, таких как ключевые слова и типы. Кроме того, небольшие вертикальные штриховые линии кода указывают, какие фигурные скобки соответствуют друг другу, а номера строк помогут вам найти нужный код позже. Чтобы свернуть или развернуть блоки кода, используйте небольшие рамки со знаком минус. Эта функция структурирования кода позволяет скрыть ненужный код на экране. Файлы вашего проекта перечислены в окне Обозревателя решений, которое находится справа.

По окончанию редактирования кода его можно запустить, выбрав «Запуск без отладки» в меню «Отладка» в строке меню (Рис 1.4). Можно также нажать клавиши CTRL+F5.

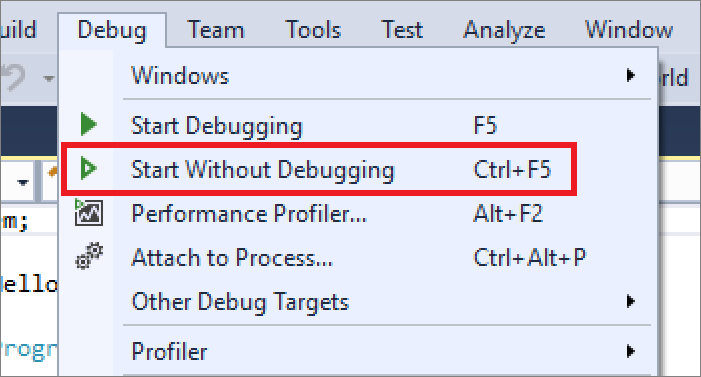


Рисунок 1.4 – Запуск программы

После нажатия среда разработки начнет компиляцию кода проекта и программа запустится, уведомив об этом пользователя. На правой стороне студии, взамен обозревателя решений появится вкладка «Средства диагностики»

Данная часть рабочего пространства дает исчерпывающую информацию о запустившейся программе, например: степень нагрузки на процессор, занятую программой оперативную память, время сеанса диагностики и т.д.

Использование окна Средства диагностики является распространенным способом профилирования приложений, но для сборок выпуска вместо этого также можно выполнять последующий анализ приложения.

[5] Средства диагностики помогают разработчикам фиксировать необходимые для запуска программы ресурсы и на их основе изменять некоторые наиболее требовательные компоненты программы для достижения наилучшей оптимизации при работе.

# 2.Разработка приложения

## 2.1.Постановка задачи

Создать Windows-приложение для обработки данных анкет. Анкету пользователь заполняет с формы приложения. В первом окне пользователь вводит свои данные. Затем появляются вопросы. Вопросы в анкете должны предполагать различные виды ответов: текстовый ответ, выбор, логический выбор, сопоставление. На форме должно быть меню, которое должно содержать следующие пункты:

1) Вывести данные на печать – результаты распечатываются (данные пользователя, все вопросы с ответами пользователя и итоговые результаты)

2) Сохранить в текстовый файл (сохранение в свободной форме).

3) Сохранить в текстовый файл по шаблону – для этого должен быть создан файл с шаблоном оформления отличным от формы, в который вставляются данные с формы в заготовленные места ввода.

4) Проверить результаты ответов на корректность заполнения.

5) Посмотреть результаты предыдущих опросов (выводятся в виде гистограммы).

6) После первого прохода пользователем анкетирования активируется пункт меню посмотреть правильные ответы.

7) Выход.

В конце анкеты должна быть кнопка «Завершить тестирование». После ее нажатия данные передаются в файл Excel в виде строки таблицы, в шапке которой все поля формы, а пользователь видит свой результат в форме. В этом файле формируется база данных опрошенных с подведением итога по тесту. После каждого 10 опрошенного формируется текстовый файл с обобщенными результатами на десять человек.

Приложение должно быть реализовано с использованием иерархии классов.

## 2.2 Обзор реализации проекта

### 2.2.1 Компоненты формы

При запуске программы появляется главная форма, на которой пользователь должен ввести необходимые данные для начала тестирования. Пользователь может выбрать кнопки «Выход» или «Посмотреть предыдущие результаты» (рис 2.1).

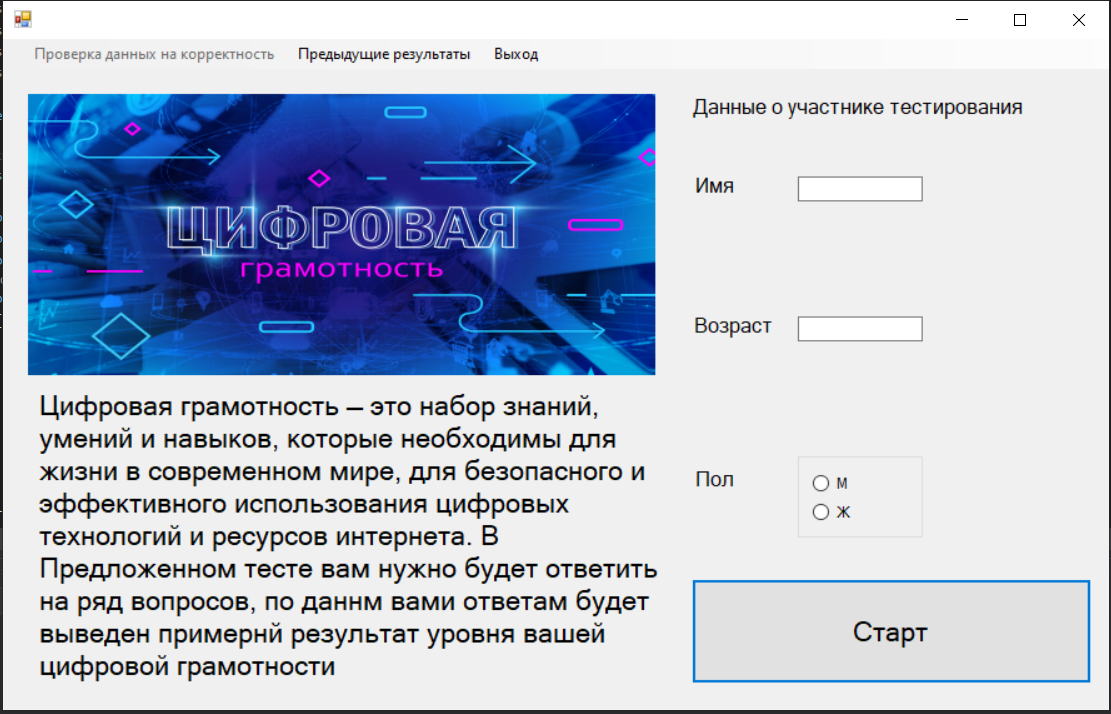


Рисунок 2.1 – Основная форма.

Кнопка «Выход» позволяет пользователю выйти из приложения (даже если тест не был пройден). Кнопка «Проверка даннх на корректность» недоступна без прохождения теста (рис 2.2).

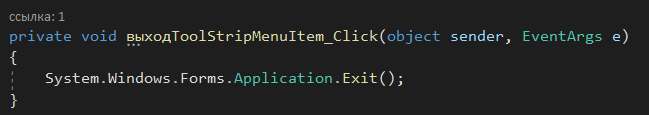


Рисунок 2.2 – Код кнопки «Выход».

Кнопка «Предыдущие результаты» выводит форму, на которой видна диаграмма с коэффициентами «ИЦГН» всех пользователей, не считая нынешнего. Форма имеет название «Statist» (рис 2.3).

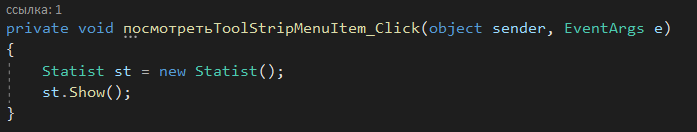


Рисунок 2.3 – Открытие формы «Statist».

Перед началом прохождения опроса пользователю необходимо заполнить информацию о себе.

При нажатии кнопки приступить к опросу открывается новая форма с вопросами «Vopros 1» (рис 2.4)

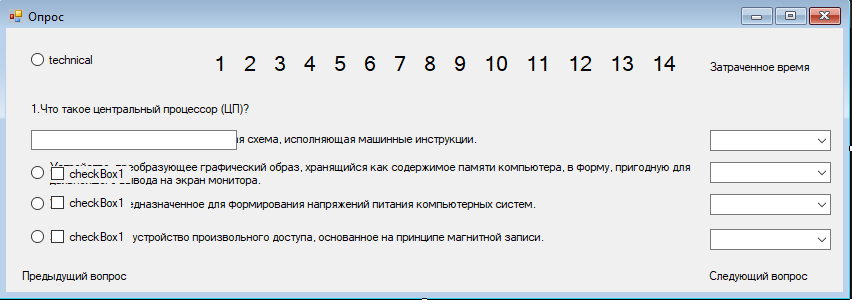


Рисунок 2.4 – Форма «Vopros 1».

Появление вопросов на данной форме осуществляется за счёт взаимодействия со свойствами таких компонентов формы, как «Label», «TextBox», «CheckBox» и «ComboBox». При инициализации формы заполняются все «ComboBox»(рис 2.5).

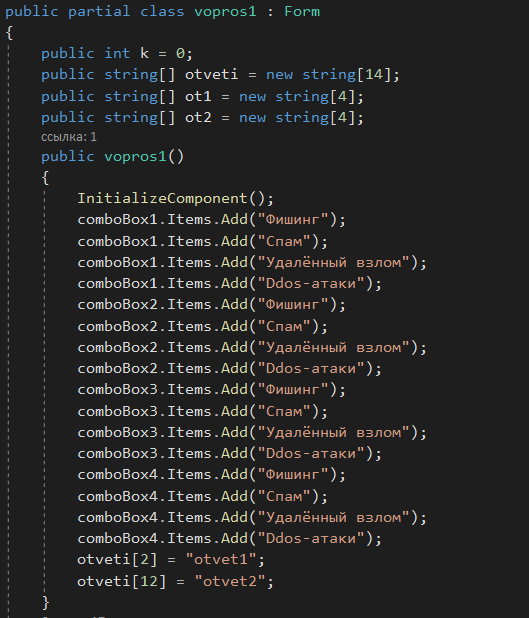


Рис 2.5 – Основные методы формы «Vopros1».

На данной форме существует три способа смены вопросов: кнопка «следующий вопрос», «Предыдущий вопрос», цифры, обозначающие номер вопроса.

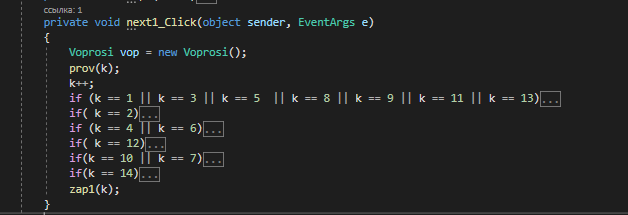


Рисунок 2.6 – Код кнопки «Следующий вопрос».

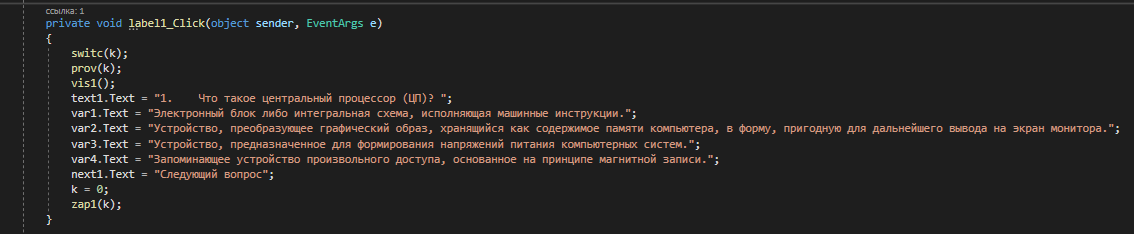


Рисунок 2.7 – Код Label1 «1».

При любом из выбранных способов перехода между вопросами цифра предыдущего вопроса закрашивается зелёным цветом, чтобы пользователю было удобнее узнавать, какие вопросы он уже просматривал.

На данной форме основной переменной является коэффициент «k», помогающий определять вопрос, на котором находится пользователь.

Пользователь может покинуть данную форму, дойдя до последнего вопроса и нажав кнопку «Завершить тестирование». При выполнении данного действия появится форма «Itog», на которой будут содержаться результаты проделанного опроса (рис 2.8).

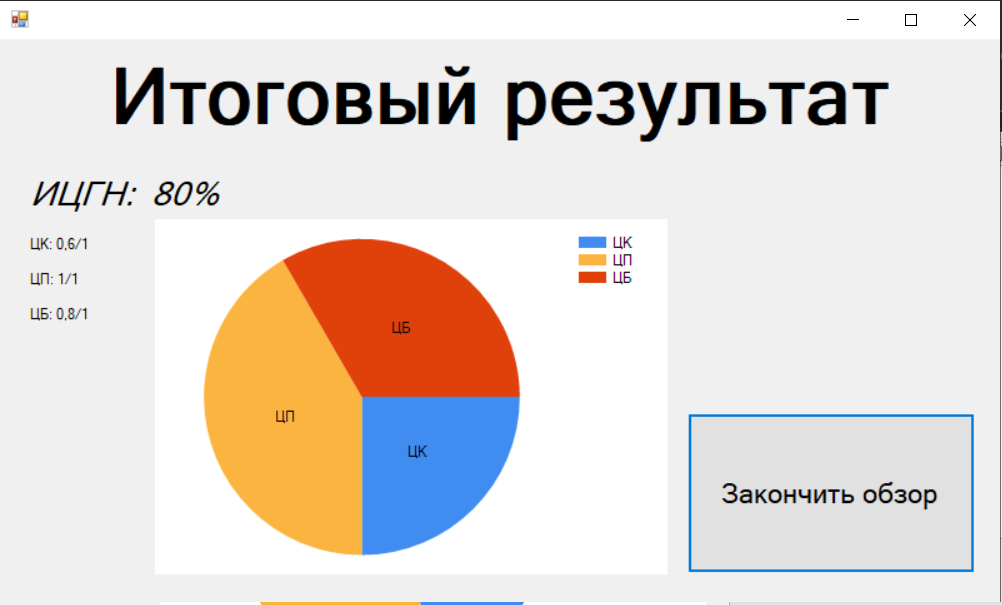


Рисунок 2.8 – Форма «Itog».

На данной форме пользователь помимо коэффициентов уровня цифровой компетенции, уровня цифрового потребления и уровня цифровой безопасности видит диаграмму с отображением всех коэффициентов. Помимо трёх коэффициентов на данной форме также находится коэффициент ИЦГН в процентах.

После нажатия кнопки «Закончить обзор» пользователь переносится обратно на форму «Form1», на которой становятся доступны ранее недоступные кнопки.

При нажатии на кнопку «Проверить на корректность» пользователь получает доступ к форме, где будут представлены все вопросы с доступными вариантами, правильные варианты ответов и выбранные пользователем ответы (Рис 2.9)

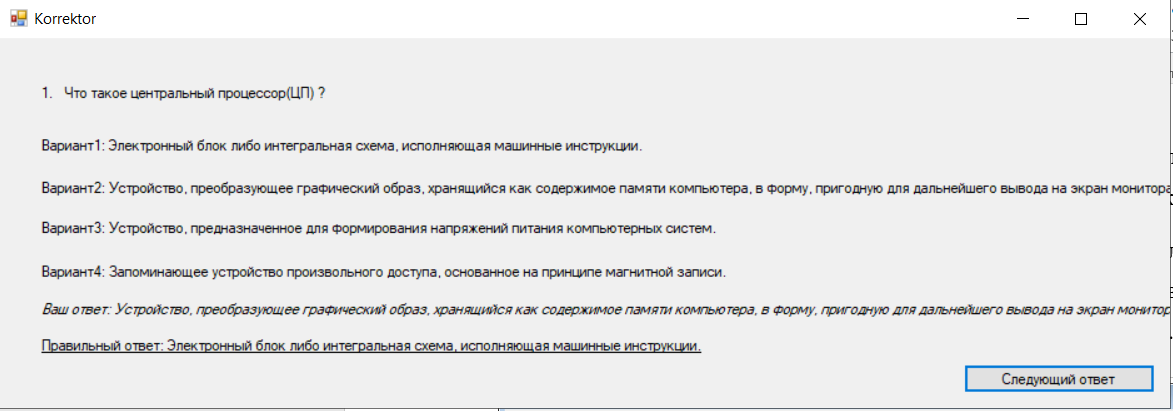


Рисунок 2.9 – Форма «Korrektor».

Пользователь на данной форме может перемещаться между вопросами с помощью нажатия на кнопку «Следующий ответ». При её нажатии как и на форме «Vopros1» меняются значения во всех элементах формы «Label»(рис 2.10).

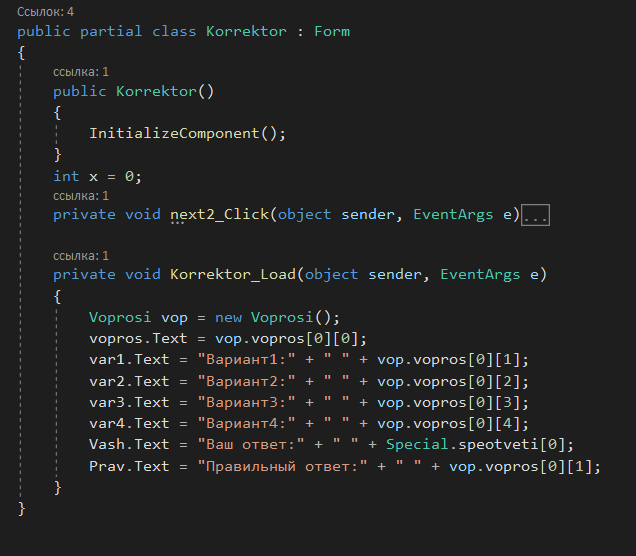


Рисунок 2.10 – Код формы «Korrektor».

Для успешного функционирования всей программы были использованы классы, которые представлены в следующем разделе «Классы»

### 2.2.2 Классы

Класс «Uchastnik» служит для сохранения и удобного использования данных о пользователе.

Класс содержит свойства (Name, Age, Sex), которые задают данные пользователя, и поля(name, age, sex), в которых хранятся данные о нём. (Рис 2.11)

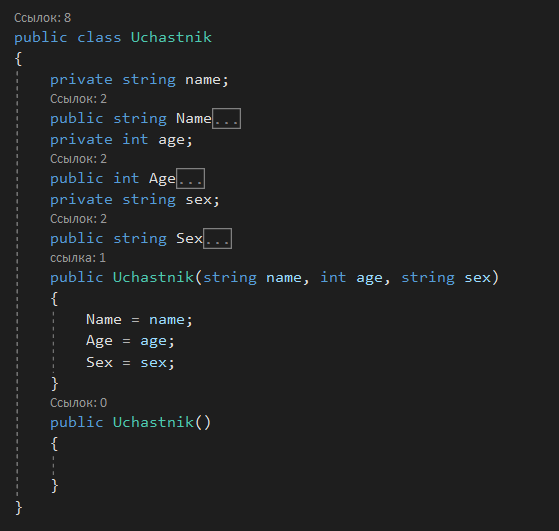


Рисунок 2.11 – Класс «Uchastnik».

В классе «Otvet» содержатся все ответы пользователя, а также преобразуется время за которое пользователь проходит тестирование (рис 2.12). В класс вносятся массив ответов из формы «Vopros1» а также переменная s2, отвечающая за кол-во времени в секундах, за которое был пройден тест. Класс ответ является наследником класса «Uchastnik».

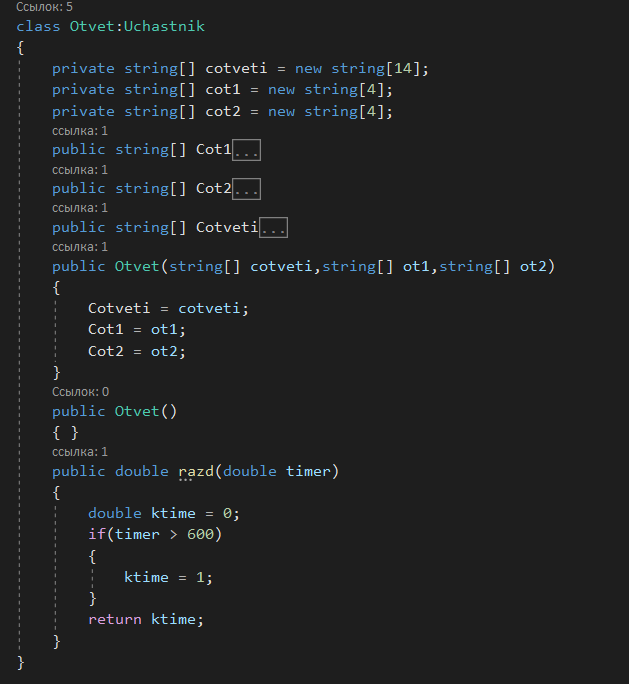


Рисунок 2.12 – код класса «Otvet».

Для обработки ответов был создан класс «CK», который преобразует их в три коэффициента по категориям «Уровень цифровых компетенций», «Уровень цифрового потребления» и «Уровень цифровой безопасности» (Рис 2.13).

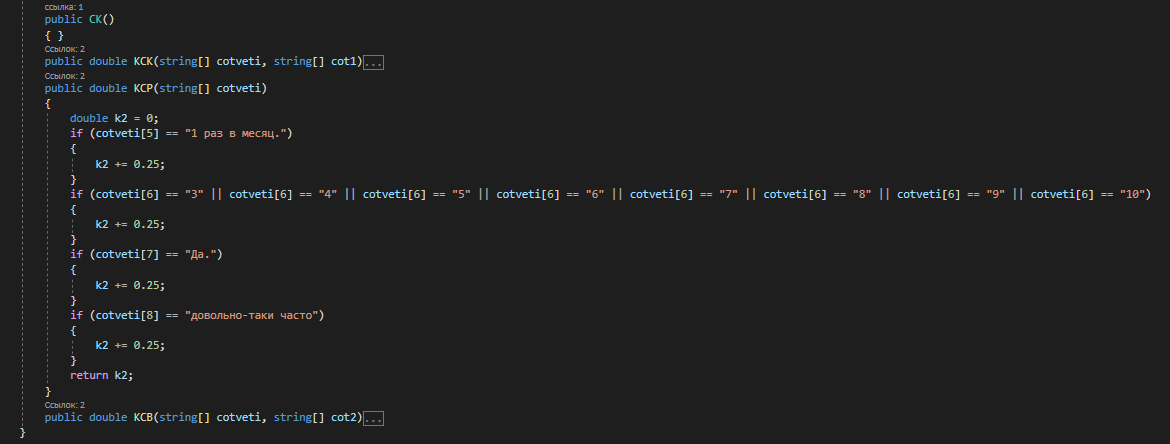


Рисунок 2.13 – код класса «CK».

В классе «CK2» обрабатываются ответы пользователя с выводом коэффициентов по каждому ответу для дальнейшего занесения их в Excel. (рис 2.14).

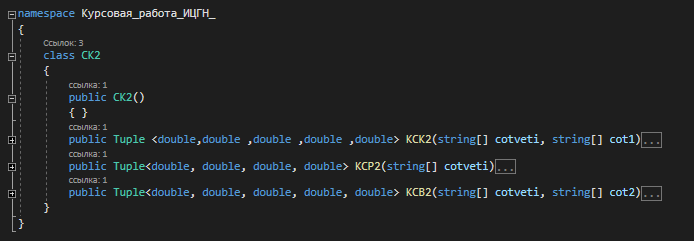


Рисунок 2.14 – код класса «CK2».

Для обработки конечного коэффициента в программе был реализован класс «ICGN». Он является наследником класса «CK» и наследует не только поля, но и методы. (Рис 2.15).

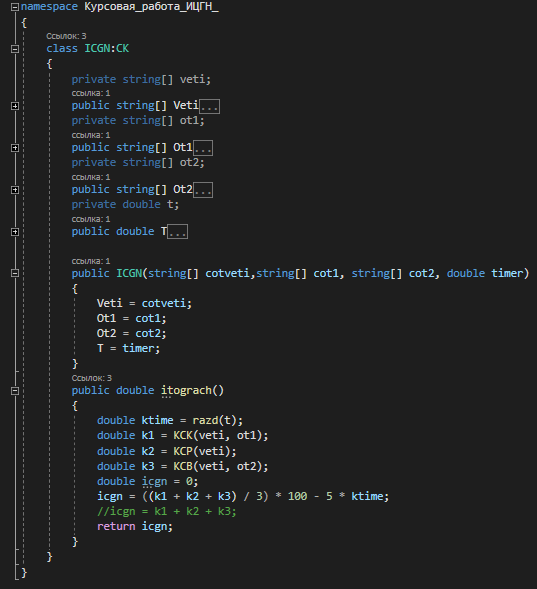


Рисунок 2.15 – Класс «ICGN».

Для работы с Excel был реализован класс, в котором были расписаны основные способы взаимодействия с ним. Был создан класс «Excel», в котором присутствуют методы для сохранения, закрытия, записывания в ячейки, считывания с ячеек, а также присутствует создания экземпляра Excel(рис 2.16).

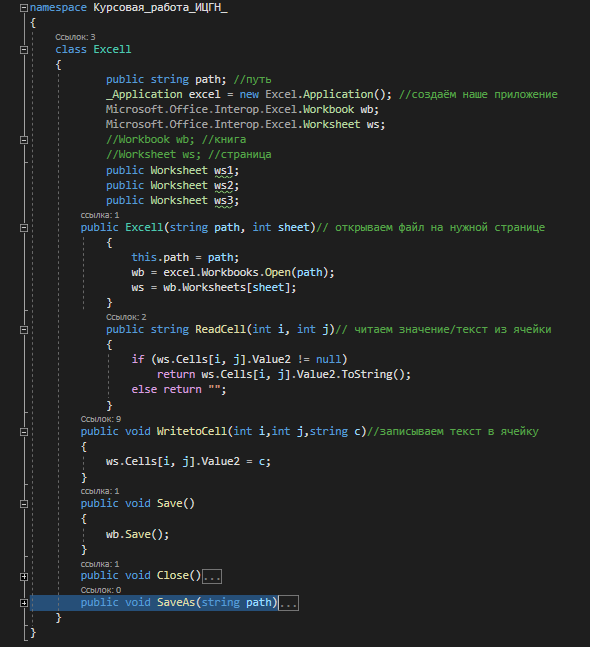


Рисунок 2.16 – Работа с Excel.

Класс Special позволяет работать с ответами и данными пользователя с любой формы. В классе присутствуют поля «specotveti», «specot1», «specot2», «specicgn» и массив классов «specPeople». Класс представлен на рисунке 2.17.

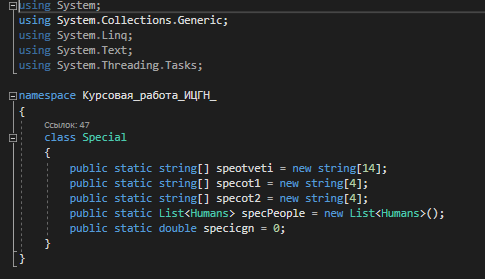
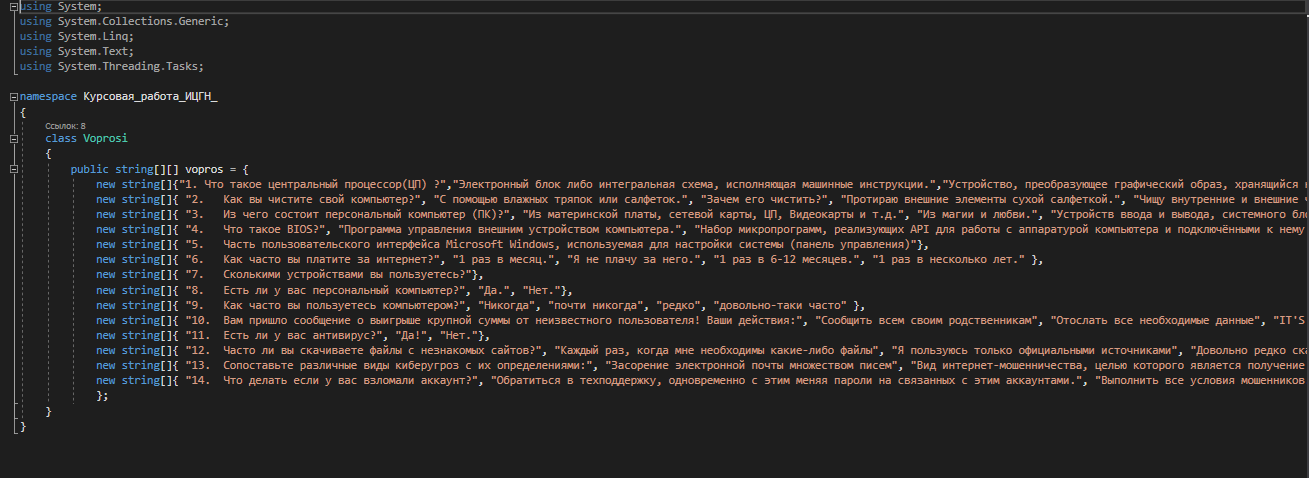


Рисунок 2.17 – Класс для хранения данных «Special».

Для удобного взаимодействия с формой «vopros1» был создан класс Voprosi, в котором хранятся все созданные вопросы (рис 2.18).

Рисунок 2.18 – Класс «Voprosi».

Иерархия классов является удобным инструментом для экономии времени при создании новых классов. Диаграмма с такой иерархией представлена на рисунке 2.19.

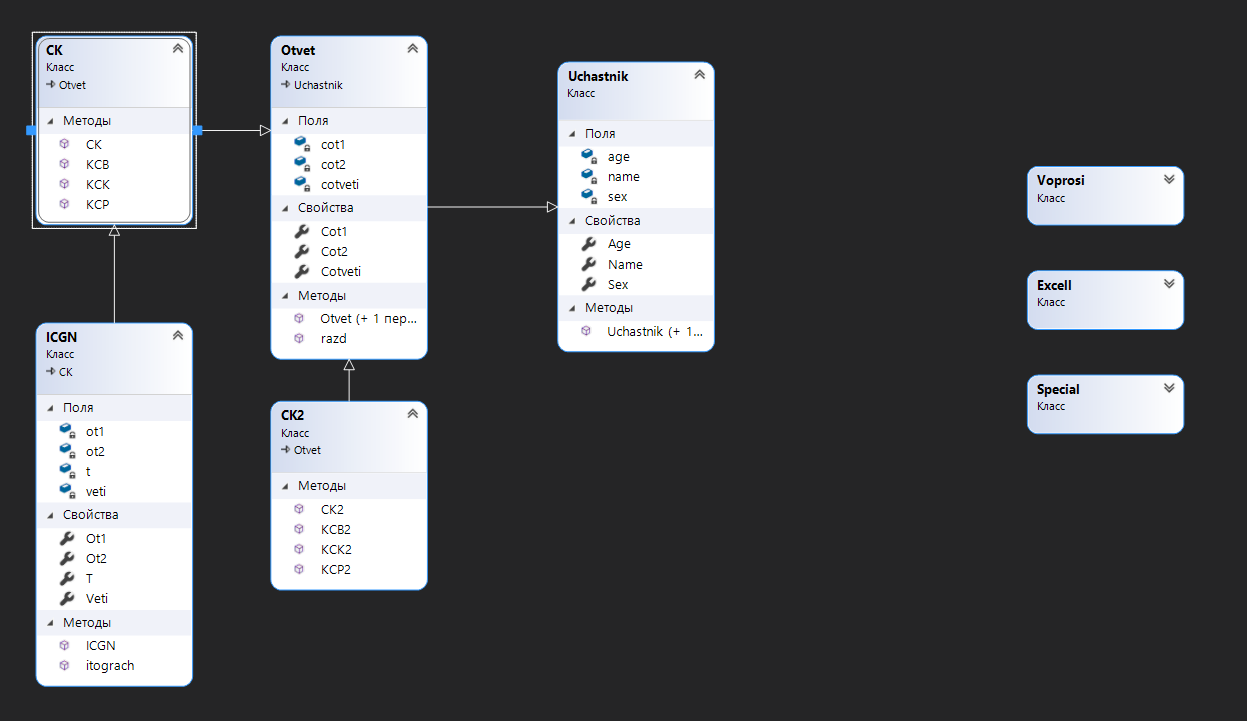


Рисунок 2.19– UML диаграмма классов

На данной диаграмме изображена вся иерархия классов. Как видно из диаграммы класс Otvet является наследником класса Uchastnik. Класс CK является наследником класс Otvet, как и класс CK2. Класс ICGN является наследником класса CK. Классы в данной иерархии наследуют поля со свойствами и методы.

# 2.3.Руководство пользователя

Запуск программы осуществляется нажатием кнопки пуск в верхней панели. Сразу после запуска на экране появится основное окно (рис 2.20).

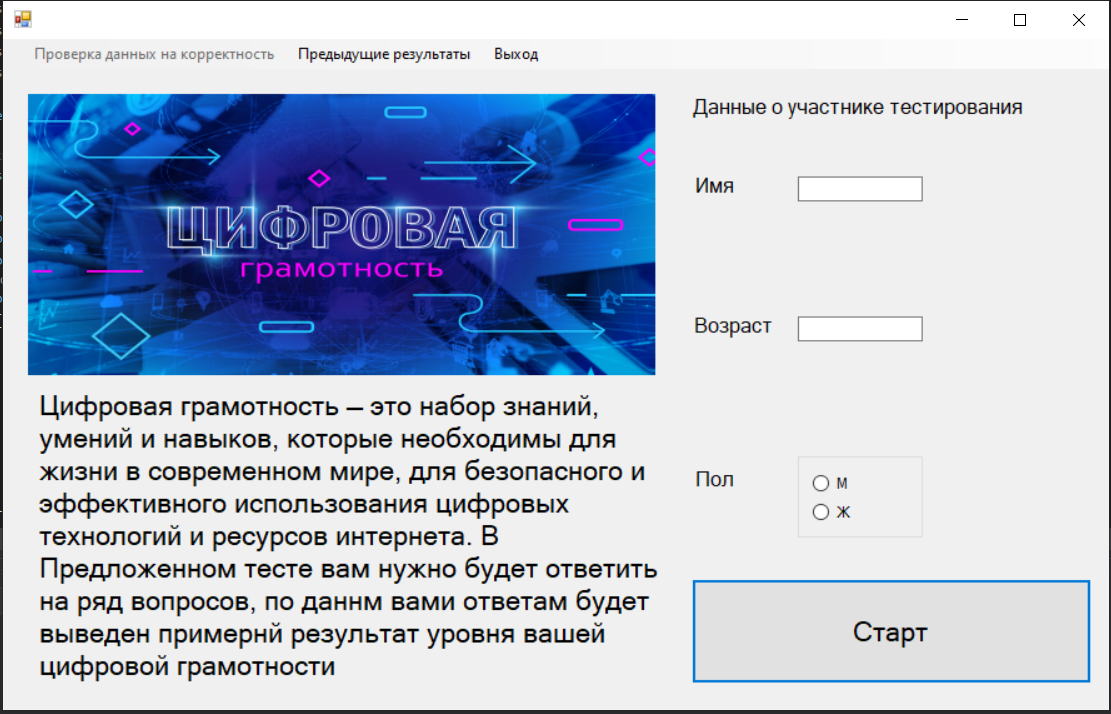


Рисунок 2.20 – Основное окно.

После запуска программы пользователь должен заполнить необходимые поля для принятия участия в опросе. Слева представлена информация о цифровой грамотности. Если пользователь не заполнит некоторые поля, программа автоматически напомнит ему это сделать.

При нажатии на кнопку «Старт» пользователь сможет приступить к заполнению теста. (Рис 2.21)

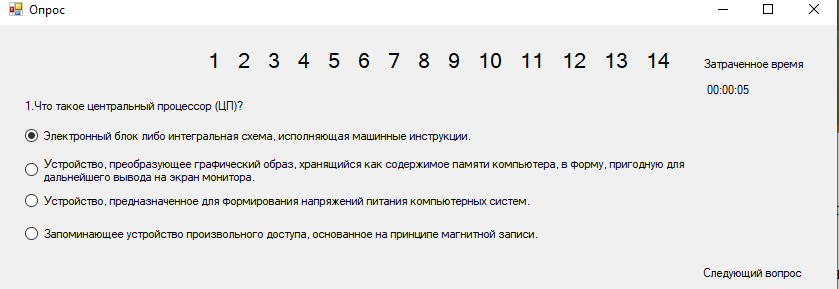


Рисунок 2.21 – Начало опроса.

При нажатии на кнопку «завершить тестирование» пользователь попадает на форму, где ему показываются его результаты.

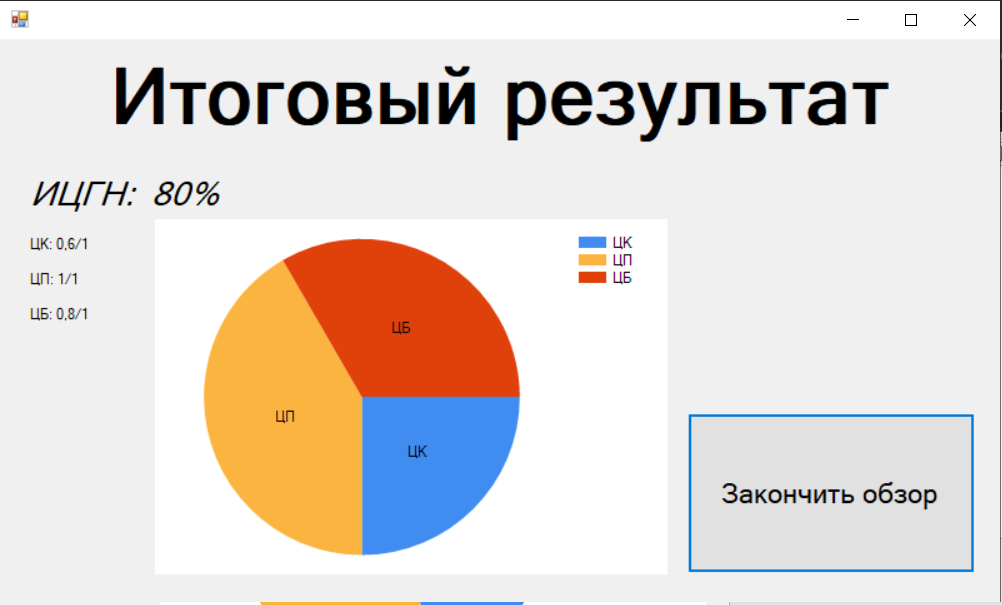


Рисунок 2.22 –Просмотр итога и выход из формы.

При нажатии на кнопку «Завершить обзор» пользователь попадёт на первоначальную форму, где будут доступны новые кнопки (рис 2.23).

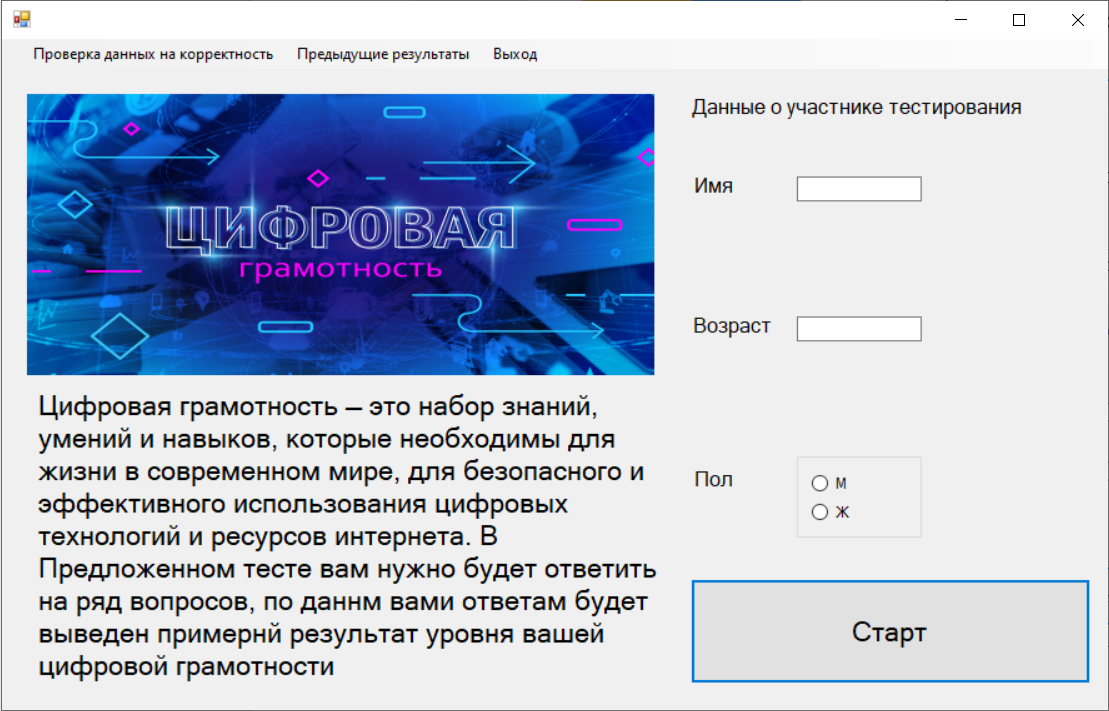


Рисунок 2.23 – Изменённая первоначальная форма.

Если пользователь захочет узнать, какие он допустил ошибки, ему необходимо будет нажать на кнопку «Проверка данных на корректность». Далее откроется форма с отображением всех вопросов по очереди. Для смены вопроса нужно будет нажать на кнопку «Следующий ответ» (рис 2.24). В конце пользователю необходимо будет нажать на кнопку «Завершить просмотр», чтобы вернуться на главную форму.

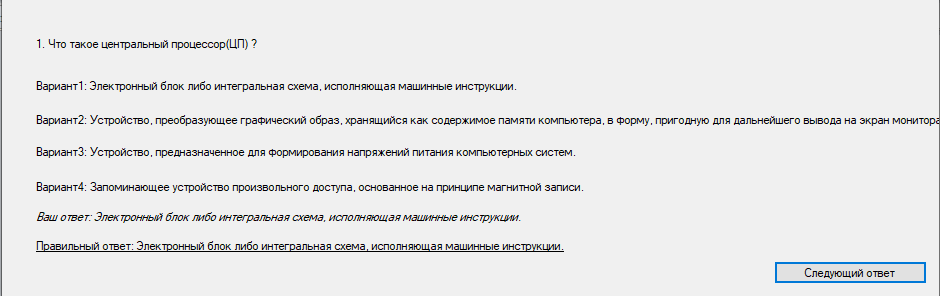


Рисунок 2.24 – Проверка на корректность.

Для того чтобы просмотреть диаграмму коэффициентов ИЦГН всех участников данного опроса необходимо нажать на кнопку «Посмотреть предыдущие результаты», и выбрать файл который показан на рисунке 2.25.

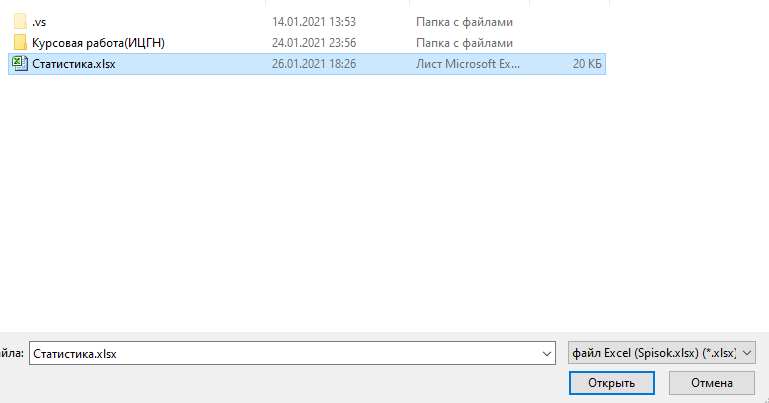


Рисунок 2.25 – Выбор файла для вывода статистики

Для выхода из программы пользователю необходимо будет нажать кнопку «Выход».

# Заключение

Во время выполнения данной курсовой работы была поставлена и выполнена задача - разработать пользовательское Windows-приложение на языке C#.

Для выполнения поставленной задачи была спроецирована иерархия классов, созданы формы со всеми нужными элементами и разработан код программы. Были составлены подходящие вопросы, а также созданы классы для обработки ответов. Были использованы дополнительные библиотеки для взаимодействия с Word и Excel. Учтены многие ошибки, которые мог бы совершить пользователь при работе с программой.

# Список литературы

1. Хейлсберг А. Язык программирования C#. Классика Computers Science [Текст]: учебное пособие / А. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд – СПб.: Питер, 2012. – 784 с.
2. Стиллмен Э. Изучаем C# [Текст]: учебное пособие / Э. Стиллмен, Д. Грин – СПб.: Питер, 2012. – 689 с.
3. Документация по C# [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения 19.01.2021)
4. Скит Д. C# для профессионалов: тонкости программирования [Текст]: учебное пособие / Д. Скит – Москва: Вильямс, 2014. – 608 с.
5. Нейгел К. C# 5.0 и платформа .NET 4.5 для профессионалов [Текст]: учебное пособие / К. Нейгел – Лос-Анджелес: Диалектика, 2013. – 816 с.
6. Общие сведения о Windows Forms [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/desktop/winfo.. (дата обращения 19.01.2021).

# Приложение А

(Обязательное)

Класс «Form1»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Microsoft.Office.Interop.Word;

using Range = Microsoft.Office.Interop.Word.Range;

using System.Drawing.Printing;

namespace Курсовая\_работа\_ИЦГН\_

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public List<Uchastnik> People = new List<Uchastnik>();

private string result = "";

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

{

if (name.Text == "" || age.Text == "" || (M.Checked == false && W.Checked == false))

{

MessageBox.Show("Пожалуйста, заполните данные о себе");

}

else

{

if(Convert.ToInt32(age.Text) <= 5)

{

MessageBox.Show("Вы слишком молоды, чтобы проходить этот тест");

}

else

{

People.Clear();

string n = name.Text;

int a = Convert.ToInt32(age.Text);

bool s = M.Checked;

string s3;

if (s == true)

{

s3 = "М";

}

else

{

s3 = "Ж";

}

People.Add(new Uchastnik(n, a, s3));

Special.specPeople = People;

vopros1 v1 = new vopros1();

v1.Show();

this.Hide();

}

}

}

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

{

System.Windows.Forms.Application.Exit();

}

private void age\_KeyPress\_1(object sender, KeyPressEventArgs e)

{

char num = e.KeyChar;

if (!Char.IsDigit(num) && num != 8) e.Handled = true;

}

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

{

char l = e.KeyChar;

if ((l < 'А' || l > 'я') && (l < 'A' || l > 'z') && l != '\b' && l != '.' && l != '-')

{

e.Handled = true;

}

}

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

{

Statist st = new Statist();

st.Show();

}

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

{

Korrektor kor = new Korrektor();

kor.Show();

this.Hide();

}

}

}

# Приложение Б

(Обязательное)

Класс «Statist»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Microsoft.Office.Interop.Excel;

using Excel = Microsoft.Office.Interop.Excel;

namespace Курсовая\_работа\_ИЦГН\_

{

public partial class Statist : Form

{

public Statist()

{

InitializeComponent();

}

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

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.DefaultExt = "\*.xls;\*.xlsx";

ofd.Filter = "файл Excel (Spisok.xlsx)|\*.xlsx";

ofd.Title = "Выберите файл базы данных";

if (ofd.ShowDialog() == DialogResult.Cancel)

{

return;

}

Excel.Application kl = new Excel.Application();

Excel.Workbook klBook = kl.Workbooks.Open(ofd.FileName,

Type.Missing, true, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing);

Excel.Worksheet klSheet = (Excel.Worksheet)klBook.Sheets[1];

Excel.Worksheet klSheet2 = (Excel.Worksheet)klBook.Sheets[2];

int i = 2;

double z = 0;

while (klSheet.Cells[i, 21].Text != "")

{

z = Convert.ToDouble(klSheet.Cells[i, 21].Value.ToString());

chart1.Series[0].Points.AddXY(i-1,z);

i++;

}

kl.Visible = true;

kl.UserControl = true;

klSheet = null;

klBook.Close(false, false, false);

kl.Quit();

klBook = null;

kl = null;

GC.Collect();

}

}

}

# Приложение В

(обязательное)

Класс «vopros1»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Курсовая\_работа\_ИЦГН\_

{

public partial class vopros1 : Form

{

public int k = 0;

public string[] otveti = new string[14];

public string[] ot1 = new string[4];

public string[] ot2 = new string[4];

public vopros1()

{

InitializeComponent();

comboBox1.Items.Add("Фишинг");

comboBox1.Items.Add("Спам");

comboBox1.Items.Add("Удалённый взлом");

comboBox1.Items.Add("Ddos-атаки");

comboBox2.Items.Add("Фишинг");

comboBox2.Items.Add("Спам");

comboBox2.Items.Add("Удалённый взлом");

comboBox2.Items.Add("Ddos-атаки");

comboBox3.Items.Add("Фишинг");

comboBox3.Items.Add("Спам");

comboBox3.Items.Add("Удалённый взлом");

comboBox3.Items.Add("Ddos-атаки");

comboBox4.Items.Add("Фишинг");

comboBox4.Items.Add("Спам");

comboBox4.Items.Add("Удалённый взлом");

comboBox4.Items.Add("Ddos-атаки");

otveti[2] = "otvet1";

otveti[12] = "otvet2";

}

private void prov(int b)

{

if (b==0 || b == 1 || b == 3 || b == 5 || b == 8 || b == 9 || b == 11 || b == 13)

{

if (var1.Checked == true)

{

otveti[b] = var1.Text;

}

if (var2.Checked == true)

{

otveti[b] = var2.Text;

}

if (var3.Checked == true)

{

otveti[b] = var3.Text;

}

if (var4.Checked == true)

{

otveti[b] = var4.Text;

}

}

if (b == 2)

{

if (logvar1.Checked == true)

{

ot1[0] = logvar1.Text;

}

if (logvar2.Checked == true)

{

ot1[1] = logvar2.Text;

}

if (logvar3.Checked == true)

{

ot1[2] = logvar3.Text;

}

if (logvar4.Checked == true)

{

ot1[3] = logvar4.Text;

}

}

if (b == 4 || b == 6)

{

if (vvod1.Text == "")

{

otveti[b] = " ";

}

else

{

otveti[b] = vvod1.Text;

}

}

if (b == 12)

{

if (comboBox1.Text != "")

{

ot2[0] = comboBox1.Text;

}

if (comboBox2.Text != "")

{

ot2[1] = comboBox2.Text;

}

if (comboBox3.Text != "")

{

ot2[2] = comboBox3.Text;

}

if (comboBox4.Text != "")

{

ot2[3] = comboBox4.Text;

}

}

if (b == 10 || b == 7)

{

if (var1.Checked == true)

{

otveti[b] = var1.Text;

}

if (var2.Checked == true)

{

otveti[b] = var2.Text;

}

}

}

private void switc(int h)

{

switch (h)

{

case 0:

label1.BackColor = Color.Green;

break;

case 1:

label2.BackColor = Color.Green;

break;

case 2:

label3.BackColor = Color.Green;

break;

case 3:

label4.BackColor = Color.Green;

break;

case 4:

label5.BackColor = Color.Green;

break;

case 5:

label6.BackColor = Color.Green;

break;

case 6:

label7.BackColor = Color.Green;

break;

case 7:

label8.BackColor = Color.Green;

break;

case 8:

label9.BackColor = Color.Green;

break;

case 9:

label10.BackColor = Color.Green;

break;

case 10:

label11.BackColor = Color.Green;

break;

case 11:

label12.BackColor = Color.Green;

break;

case 12:

label13.BackColor = Color.Green;

break;

case 13:

label14.BackColor = Color.Green;

break;

}

}

private void vis1()

{

var1.Visible = true;

var2.Visible = true;

var3.Visible = true;

var4.Visible = true;

comboBox1.Visible = false;

comboBox2.Visible = false;

comboBox3.Visible = false;

comboBox4.Visible = false;

vvod1.Visible = false;

logvar1.Visible = false;

logvar2.Visible = false;

logvar3.Visible = false;

logvar4.Visible = false;

back1.Visible = true;

}

private void vis2()

{

var1.Visible = false;

var2.Visible = false;

var3.Visible = false;

var4.Visible = false;

comboBox1.Visible = false;

comboBox2.Visible = false;

comboBox3.Visible = false;

comboBox4.Visible = false;

vvod1.Visible = false;

logvar1.Visible = true;

logvar2.Visible = true;

logvar3.Visible = true;

logvar4.Visible = true;

back1.Visible = true;

}

private void vis3()

{

var1.Visible = false;

var2.Visible = false;

var3.Visible = false;

var4.Visible = false;

comboBox1.Visible = false;

comboBox2.Visible = false;

comboBox3.Visible = false;

comboBox4.Visible = false;

vvod1.Visible = true;

logvar1.Visible = false;

logvar2.Visible = false;

logvar3.Visible = false;

logvar4.Visible = false;

back1.Visible = true;

}

private void vis4()

{

var1.Visible = true;

var2.Visible = true;

var3.Visible = false;

var4.Visible = false;

comboBox1.Visible = false;

comboBox2.Visible = false;

comboBox3.Visible = false;

comboBox4.Visible = false;

vvod1.Visible = false;

logvar1.Visible = false;

logvar2.Visible = false;

logvar3.Visible = false;

logvar4.Visible = false;

back1.Visible = true;

}

private void vis5()

{

var1.Visible = true;

var2.Visible = true;

var3.Visible = true;

var4.Visible = true;

comboBox1.Visible = true;

comboBox2.Visible = true;

comboBox3.Visible = true;

comboBox4.Visible = true;

vvod1.Visible = false;

logvar1.Visible = false;

logvar2.Visible = false;

logvar3.Visible = false;

logvar4.Visible = false;

back1.Visible = true;

}

private void zap1(int d)

{

if(d == 0 || d == 1 || d == 3 || d == 5 || d == 8 || d == 9 || d == 11 || d == 13)

{

if(otveti[d] == var1.Text)

{

var1.Checked = true;

}

if(otveti[d] == var2.Text)

{

var2.Checked = true;

}

if (otveti[d] == var3.Text)

{

var3.Checked = true;

}

if (otveti[d] == var4.Text)

{

var4.Checked = true;

}

}

if(d == 2)

{

if(otveti[d] != "")

{

if (ot1[0] == logvar1.Text)

{

logvar1.Checked = true;

}

if (ot1[1] == logvar2.Text)

{

logvar2.Checked = true;

}

if (ot1[2] == logvar3.Text)

{

logvar3.Checked = true;

}

if (ot1[3] == logvar4.Text)

{

logvar4.Checked = true;

}

}

}

if(d == 4 || d == 6)

{

vvod1.Text = otveti[d];

}

if(d == 12)

{

if(otveti[d] != "")

{

comboBox1.Text = ot2[0];

comboBox2.Text = ot2[1];

comboBox3.Text = ot2[2];

comboBox4.Text = ot2[3];

}

}

if(d == 7 || d == 10)

{

if (otveti[d] == var1.Text)

{

var1.Checked = true;

}

if (otveti[d] == var2.Text)

{

var2.Checked = true;

}

}

}

private void next1\_Click(object sender, EventArgs e)

{

Voprosi vop = new Voprosi();

prov(k);

k++;

if (k == 1 || k == 3 || k == 5 || k == 8 || k == 9 || k == 11 || k == 13)

{

text1.Text = vop.vopros[k][0];

var1.Text = vop.vopros[k][1];

var2.Text = vop.vopros[k][2];

var3.Text = vop.vopros[k][3];

var4.Text = vop.vopros[k][4];

vis1();

if (k == 1)

{

label1.BackColor = Color.Green;

}

if (k == 3)

{

label3.BackColor = Color.Green;

}

if (k == 5)

{

label5.BackColor = Color.Green;

}

if (k == 8)

{

label8.BackColor = Color.Green;

}

if (k == 9)

{

label9.BackColor = Color.Green;

}

if (k == 11)

{

label11.BackColor = Color.Green;

}

if (k == 13)

{

next1.Text = "Завершить тестирование";

label13.BackColor = Color.Green;

}

}

if( k == 2)

{

text1.Text = vop.vopros[k][0];

logvar1.Text = vop.vopros[k][1];

logvar2.Text = vop.vopros[k][2];

logvar3.Text = vop.vopros[k][3];

logvar4.Text = vop.vopros[k][4];

vis2();

label2.BackColor = Color.Green;

otveti[k] = "";

}

if (k == 4 || k == 6)

{

text1.Text = vop.vopros[k][0];

vis3();

if (k == 4)

{

label4.BackColor = Color.Green;

}

if (k == 6)

{

label6.BackColor = Color.Green;

}

}

if( k == 12)

{

text1.Text = vop.vopros[k][0];

var1.Text = vop.vopros[k][1];

var2.Text = vop.vopros[k][2];

var3.Text = vop.vopros[k][3];

var4.Text = vop.vopros[k][4];

vis5();

label12.BackColor = Color.Green;

}

if(k == 10 || k == 7)

{

text1.Text = vop.vopros[k][0];

var1.Text = vop.vopros[k][1];

var2.Text = vop.vopros[k][2];

vis4();

if (k == 10)

{

label10.BackColor = Color.Green;

}

if (k == 7)

{

label7.BackColor = Color.Green;

}

}

if(k == 14)

{

prov(k - 1);

label14.BackColor = Color.Green;

new Otvet(otveti,ot1,ot2);

Itog t = new Itog();

t.Owner = this;

timer1.Stop();

t.Show();

this.Hide();

}

zap1(k);

}

private void label1\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis1();

text1.Text = "1. Что такое центральный процессор (ЦП)? ";

var1.Text = "Электронный блок либо интегральная схема, исполняющая машинные инструкции.";

var2.Text = "Устройство, преобразующее графический образ, хранящийся как содержимое памяти компьютера, в форму, пригодную для дальнейшего вывода на экран монитора.";

var3.Text = "Устройство, предназначенное для формирования напряжений питания компьютерных систем.";

var4.Text = "Запоминающее устройство произвольного доступа, основанное на принципе магнитной записи.";

next1.Text = "Следующий вопрос";

k = 0;

zap1(k);

}

private void label2\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis1();

text1.Text = "2. Как вы чистите свой компьютер?";

var1.Text = "С помощью влажных тряпок или салфеток.";

var2.Text = "Зачем его чистить?";

var3.Text = "Протираю внешние элементы сухой салфеткой.";

var4.Text = "Чищу внутренние и внешние части компьютера с помощью специальных чистящих средств.";

next1.Text = "Следующий вопрос";

k = 1;

zap1(k);

}

private void label3\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

text1.Text = "3. Из чего состоит персональный компьютер (ПК)? ";

vis2();

logvar1.Text = "Из материнской платы, сетевой карты, ЦП, Видеокарты и т.д.";

logvar2.Text = "Из магии и любви.";

logvar3.Text = "Устройств ввода и вывода, системного блока.";

logvar4.Text = "из микросхем и магии.";

next1.Text = "Следующий вопрос";

k = 2;

zap1(k);

}

private void label4\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

text1.Text = "4. Что такое BIOS? ";

vis1();

var1.Text = "Программа управления внешним устройством компьютера.";

var2.Text = "Набор микропрограмм, реализующих API для работы с аппаратурой компьютера и подключёнными к нему устройствами.";

var3.Text = "Основной накопителель данных в большинстве компьютеров.";

var4.Text = "Электронные и механические части вычислительного устройства.";

next1.Text = "Следующий вопрос";

k = 3;

zap1(k);

}

private void label5\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

text1.Text = "5. Часть пользовательского интерфейса Microsoft Windows, используемая для настройки системы (панель управления) ";

vis3();

next1.Text = "Следующий вопрос";

k = 4;

zap1(k);

}

private void label6\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis1();

text1.Text = "6. Как часто вы платите за интернет?";

var1.Text = "1 раз в месяц.";

var2.Text = "Я не плачу за него.";

var3.Text = "1 раз в 6-12 месяцев.";

var4.Text = "1 раз в несколько лет.";

next1.Text = "Следующий вопрос";

k = 5;

zap1(k);

}

private void label7\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

text1.Text = "7. Сколькими устройствами вы пользуетесь?";

next1.Text = "Следующий вопрос";

vis3();

k = 6;

zap1(k);

}

private void label8\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

text1.Text = "8. Есть ли у вас персональный компьютер? ";

next1.Text = "Следующий вопрос";

vis4();

var1.Text = "Да.";

var2.Text = "Нет.";

k = 7;

zap1(k);

}

private void label9\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis1();

text1.Text = "9. Как часто вы пользуетесь компьютером? ";

next1.Text = "Следующий вопрос";

var1.Text = "Никогда";

var2.Text = "почти никогда";

var3.Text = "редко";

var4.Text = "довольно-таки часто";

k = 8;

zap1(k);

}

private void label10\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis1();

text1.Text = "10. Вам пришло сообщение о выигрыше крупной суммы от неизвестного пользователя! Ваши действия: ";

next1.Text = "Следующий вопрос";

var1.Text = "Сообщить всем своим родственникам";

var2.Text = "Отослать все необходимые данные";

var3.Text = "IT'S A TRAP!";

var4.Text = "Не стану ничего высылать, проигнорировав сообщение";

k = 9;

zap1(k);

}

private void label11\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis4();

text1.Text = "11. Есть ли у вас антивирус? ";

next1.Text = "Следующий вопрос";

var1.Text = "Да!";

var2.Text = "Нет.";

k = 10;

zap1(k);

}

private void label12\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis1();

text1.Text = "12. Часто ли вы скачиваете файлы с незнакомых сайтов?";

var1.Text = "Каждый раз, когда мне необходимы какие-либо файлы";

var2.Text = "Я пользуюсь только официальными источниками";

var3.Text = "Довольно редко скачиваю";

var4.Text = "Стараюсь не скачивать файлы с незнакомых сайтов";

next1.Text = "Следующий вопрос";

k = 11;

zap1(k);

}

private void label13\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis5();

text1.Text = "13. Сопоставьте различные виды киберугроз с их определениями: ";

var1.Text = "Засорение электронной почты множеством писем";

var2.Text = "Вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей";

var3.Text = "Взлом ПК из любой точки мира";

var4.Text = "Специальным образом организованные массированные сетевые запросы";

next1.Text = "Следующий вопрос";

k = 12;

zap1(k);

}

private void label14\_Click(object sender, EventArgs e)

{

switc(k);

prov(k);

vis1();

text1.Text = "14. Что делать если у вас взломали аккаунт?";

var1.Text = "Обратиться в техподдержку, одновременно с этим меняя пароли на связанных с этим аккаунтами.";

var2.Text = "Выполнить все условия мошенников.";

var3.Text = "Игнорировать взлом.";

var4.Text = "Удалить свой аккаунт.";

next1.Text = "Завершить тестирование";

k = 13;

zap1(k);

}

private void back1\_Click(object sender, EventArgs e)

{

prov(k);

k--;

Voprosi vop = new Voprosi();

text1.Text = vop.vopros[k][0];

if (k == 0 || k == 1 || k == 3 || k == 5 || k == 8 || k == 11 || k == 9)

{

var1.Text = vop.vopros[k][1];

var2.Text = vop.vopros[k][2];

var3.Text = vop.vopros[k][3];

var4.Text = vop.vopros[k][4];

vis1();

next1.Text = "Следующий вопрос";

if (k == 9)

{

label11.BackColor = Color.Green;

}

if (k == 11)

{

label13.BackColor = Color.Green;

}

if (k == 8)

{

label10.BackColor = Color.Green;

}

if (k == 5)

{

label7.BackColor = Color.Green;

}

if (k == 3)

{

label5.BackColor = Color.Green;

}

if (k == 1)

{

label3.BackColor = Color.Green;

}

if (k == 0)

{

back1.Visible = false;

label2.BackColor = Color.Green;

}

}

if (k == 2)

{

logvar1.Text = vop.vopros[k][1];

logvar2.Text = vop.vopros[k][2];

logvar3.Text = vop.vopros[k][3];

logvar4.Text = vop.vopros[k][4];

vis2();

next1.Text = "Следующий вопрос";

label4.BackColor = Color.Green;

}

if (k == 4 || k == 6)

{

vis3();

next1.Text = "Следующий вопрос";

if (k == 4)

{

label6.BackColor = Color.Green;

}

if (k == 6)

{

label8.BackColor = Color.Green;

}

}

if (k == 12)

{

var1.Text = vop.vopros[k][1];

var2.Text = vop.vopros[k][2];

var3.Text = vop.vopros[k][3];

var4.Text = vop.vopros[k][4];

vis5();

label14.BackColor = Color.Green;

next1.Text = "Следующий вопрос";

}

if (k == 10 || k == 7)

{

var1.Text = vop.vopros[k][1];

var2.Text = vop.vopros[k][2];

vis4();

next1.Text = "Следующий вопрос";

if (k == 7)

{

label9.BackColor = Color.Green;

}

if (k == 10)

{

label12.BackColor = Color.Green;

}

}

zap1(k);

}

private void vopros1\_Load(object sender, EventArgs e)

{

timer1.Start();

}

int h = 0;

int m = 0;

int s = 0;

public double s2 = 0;

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

{

s2++;

s++;

string time = "";

if (s > 59)

{

m++;

s = 0;

}

if (m > 59)

{

h++;

m = 0;

}

if (h < 10)

{

time += "0" + h;

}

else

{

time += h;

}

time += ":";

if (m < 10)

{

time += "0" + m;

}

else

{

time += m;

}

time += ":";

if (s < 10)

{

time += "0" + s;

}

else

{

time += s;

}

Timer.Text = time;

}

private void comboBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

char l = e.KeyChar;

if (l < 'А' || l >= 'А')

{

e.Handled = true;

}

}

private void comboBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

char l = e.KeyChar;

if (l < 'А' || l >= 'А')

{

e.Handled = true;

}

}

private void comboBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

char l = e.KeyChar;

if (l < 'А' || l >= 'А')

{

e.Handled = true;

}

}

private void comboBox4\_KeyPress(object sender, KeyPressEventArgs e)

{

char l = e.KeyChar;

if (l < 'А' || l >= 'А')

{

e.Handled = true;

}

}

//private void vopros1\_Load(object sender, EventArgs e)

//{

// timer1.Start();

//}

//DateTime date1 = new DateTime(0, 0);

//private void timer1\_Tick\_1(object sender, EventArgs e)

//{

// date1 = date1.AddSeconds(1);

// Timer.Text = date1.ToString("mm:ss");

//}

}

}

# Приложение Г

(обязательное)

Класс «Itog»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Media;

using Microsoft.Office.Interop.Excel;

using Excel = Microsoft.Office.Interop.Excel;

namespace Курсовая\_работа\_ИЦГН\_

{

public partial class Itog : Form

{

SoundPlayer sndPlayer;

public Itog()

{

InitializeComponent();

}

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

{

Form1 f = new Form1();

//f.Refresh();

f.корректностьToolStripMenuItem.Enabled = true;

f.Show();

this.Hide();

}

private void Itog\_Load\_1(object sender, EventArgs e)

{

CK p1 = new CK();

vopros1 v = this.Owner as vopros1;

//Form1 f1 = this.Owner as Form1;

ICGN I1 = new ICGN(v.otveti,v.ot1,v.ot2,v.s2);

double k1 = p1.KCK(v.otveti, v.ot1);

double k2 = p1.KCP(v.otveti);

double k3 = p1.KCB(v.otveti, v.ot2);

label3.Text = "ЦК: " + k1.ToString() + "/1";

label4.Text = "ЦП: " + k2.ToString() + "/1";

label5.Text = "ЦБ: " + k3.ToString() + "/1";

label2.Text = "ИЦГН:" + " " + Math.Round(I1.itograch()).ToString() + "%";

Special.specicgn = Math.Round(I1.itograch(),0);

chart1.Series[0].Points.AddXY("ЦК", k1);

chart1.Series[0].Points.AddXY("ЦП", k2);

chart1.Series[0].Points.AddXY("ЦБ", k3);

Special.specot1 = v.ot1;

Special.specot2 = v.ot2;

Special.speotveti = v.otveti;

int k = 2;

Excell file = new Excell("C:\\Users\\d-les\\Downloads\\Курсовая работа(ИЦГН)\\Статистика.xlsx", 1);

while (file.ReadCell(k, 1) != "")

{

k++;

}

file.WritetoCell(k, 1, Special.specPeople[0].Name);

file.WritetoCell(k, 2, (Special.specPeople[0].Age).ToString());

file.WritetoCell(k, 3, Special.specPeople[0].Sex);

double[] t = new double[14];

CK2 p2 = new CK2();

var result1 = p2.KCK2(v.otveti, v.ot1);

t[0] = result1.Item1;

t[1] = result1.Item2;

t[2] = result1.Item3;

t[3] = result1.Item4;

t[4] = result1.Item5;

var result2 = p2.KCP2(v.otveti);

t[5] = result2.Item1;

t[6] = result2.Item2;

t[7] = result2.Item3;

t[8] = result2.Item4;

var result3 = p2.KCB2(v.otveti,v.ot2);

t[9] = result3.Item1;

t[10] = result3.Item2;

t[11] = result3.Item3;

t[12] = result3.Item4;

t[13] = result3.Item5;

for (int i = 0; i < 14; i++)

{

file.WritetoCell(k, i + 4, (t[i]).ToString());

}

file.WritetoCell(k, 18, (k1).ToString());

file.WritetoCell(k, 19, (k2).ToString());

file.WritetoCell(k, 20, (k3).ToString());

file.WritetoCell(k, 21, (I1.itograch()/100).ToString());

//MessageBox.Show(file.ReadCell(21, 21));

double finalitgn = 0;

for (int m = k; m > 1; m--)

{

finalitgn += Convert.ToDouble(file.ReadCell(m, 21));

}

finalitgn = finalitgn / k;

file.WritetoCell(2, 22, (finalitgn).ToString());

file.Save();

file.Close();

}

}

}

# Приложение Д

(обязательное)

Класс «Korrektor»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Курсовая\_работа\_ИЦГН\_

{

public partial class Korrektor : Form

{

public Korrektor()

{

InitializeComponent();

}

int x = 0;

private void next2\_Click(object sender, EventArgs e)

{

Voprosi vop = new Voprosi();

x++;

if(x == 1 || x == 3 || x == 5 || x == 8 || x == 9 || x == 11 || x == 13)

{

vopros.Text = vop.vopros[x][0];

var1.Text = "Вариант1:" + " " + vop.vopros[x][1];

var2.Text = "Вариант2:" + " " + vop.vopros[x][2];

var3.Text = "Вариант3:" + " " + vop.vopros[x][3];

var4.Text = "Вариант4:" + " " + vop.vopros[x][4];

Vash.Text = "Ваш ответ:" + " " + Special.speotveti[x];

var1.Visible = true;

var2.Visible = true;

var3.Visible = true;

var4.Visible = true;

if(x == 1)

{

Prav.Text = "Правильный ответ:" + " " + "Чищу внутренние и внешние части компьютера с помощью специальных чистящих средств.";

}

if (x == 3)

{

Prav.Text = "Правильный ответ:" + " " + "Набор микропрограмм, реализующих API для работы с аппаратурой компьютера и подключёнными к нему устройствами.";

}

if (x == 5)

{

Prav.Text = "Правильный ответ:" + " " + "1 раз в месяц.";

}

if (x == 8)

{

Prav.Text = "Правильный ответ:" + " " + "довольно-таки часто";

}

if (x == 9)

{

Prav.Text = "Правильный ответ:" + " " + "IT'S A TRAP! или Не стану ничего высылать, проигнорировав сообщение";

}

if (x == 11)

{

Prav.Text = "Правильный ответ:" + " " + "Я пользуюсь только официальными источниками";

}

if (x == 13)

{

Prav.Text = "Правильный ответ:" + " " + "Обратиться в техподдержку, одновременно с этим меняя пароли на связанных с этим аккаунтами.";

next2.Text = "Завершить просмотр";

}

}

if(x == 7 || x == 10)

{

vopros.Text = vop.vopros[x][0];

var1.Text = "Вариант1:" + " " + vop.vopros[x][1];

var2.Text = "Вариант2:" + " " + vop.vopros[x][2];

Vash.Text = "Ваш ответ:" + " " + Special.speotveti[x];

var1.Visible = true;

var2.Visible = true;

var3.Visible = false;

var4.Visible = false;

if (x == 7)

{

Prav.Text = "Правильный ответ:" + " " + "Да.";

}

else

{

Prav.Text = "Правильный ответ:" + " " + "Да!";

}

}

if(x == 4)

{

vopros.Text = vop.vopros[x][0];

Prav.Text = "Правильный ответ:" + " " + "панель управления";

Vash.Text = "Ваш ответ:" + " " + Special.speotveti[x];

var1.Visible = false;

var2.Visible = false;

var3.Visible = false;

var4.Visible = false;

}

if (x == 6)

{

vopros.Text = vop.vopros[x][0];

Prav.Text = "Правильный ответ:" + " " + ">3 и <12";

Vash.Text = "Ваш ответ:" + " " + Special.speotveti[x];

var1.Visible = false;

var2.Visible = false;

var3.Visible = false;

var4.Visible = false;

}

if (x == 12)

{

vopros.Text = vop.vopros[x][0];

var1.Text = "Вариант1:" + " " + vop.vopros[x][1];

var2.Text = "Вариант2:" + " " + vop.vopros[x][2];

var3.Text = "Вариант3:" + " " + vop.vopros[x][3];

var4.Text = "Вариант4:" + " " + vop.vopros[x][4];

var1.Visible = true;

var2.Visible = true;

var3.Visible = true;

var4.Visible = true;

Prav.Text = "Правильный ответ:" + " " + "1 - Спам, 2 - Фишинг, 3 - Удалённый взлом, 4 - Ddos-атака";

Vash.Text = "Ваш ответ:" + " " + "1-" + Special.specot2[0]+ " " + "2-" + Special.specot2[1] + " " + "3-" + Special.specot2[2] + " " + "4-" + Special.specot2[3];

}

if(x == 2)

{

vopros.Text = vop.vopros[x][0];

var1.Text = "Вариант1:" + " " + vop.vopros[x][1];

var2.Text = "Вариант2:" + " " + vop.vopros[x][2];

var3.Text = "Вариант3:" + " " + vop.vopros[x][3];

var4.Text = "Вариант4:" + " " + vop.vopros[x][4];

var1.Visible = true;

var2.Visible = true;

var3.Visible = true;

var4.Visible = true;

Prav.Text = "Правильный ответ:" + " " + "Из материнской платы, сетевой карты, ЦП, Видеокарты и т.д." + " " + "Устройств ввода и вывода, системного блока.";

Vash.Text = "Ваш ответ:" + " " + Special.specot1[0] + " " + Special.specot1[2];

}

if (x == 14)

{

Form1 fo = new Form1();

fo.корректностьToolStripMenuItem.Enabled = true;

fo.Show();

this.Hide();

}

}

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

{

Voprosi vop = new Voprosi();

vopros.Text = vop.vopros[0][0];

var1.Text = "Вариант1:" + " " + vop.vopros[0][1];

var2.Text = "Вариант2:" + " " + vop.vopros[0][2];

var3.Text = "Вариант3:" + " " + vop.vopros[0][3];

var4.Text = "Вариант4:" + " " + vop.vopros[0][4];

Vash.Text = "Ваш ответ:" + " " + Special.speotveti[0];

Prav.Text = "Правильный ответ:" + " " + vop.vopros[0][1];

}

}

}

# Приложение Е

(Обязательное)

Класс «Voprosi»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Курсовая\_работа\_ИЦГН\_

{

class Voprosi

{

public string[][] vopros = {

new string[]{ "1. Что такое центральный процессор(ЦП) ?","Электронный блок либо интегральная схема, исполняющая машинные инструкции.","Устройство, преобразующее графический образ, хранящийся как содержимое памяти компьютера, в форму, пригодную для дальнейшего вывода на экран монитора.","Устройство, предназначенное для формирования напряжений питания компьютерных систем.","Запоминающее устройство произвольного доступа, основанное на принципе магнитной записи." },

new string[]{ "2. Как вы чистите свой компьютер?", "С помощью влажных тряпок или салфеток.", "Зачем его чистить?", "Протираю внешние элементы сухой салфеткой.", "Чищу внутренние и внешние части компьютера с помощью специальных чистящих средств." },

new string[]{ "3. Из чего состоит персональный компьютер (ПК)?", "Из материнской платы, сетевой карты, ЦП, Видеокарты и т.д.", "Из магии и любви.", "Устройств ввода и вывода, системного блока.", "из микросхем и магии." },

new string[]{ "4. Что такое BIOS?", "Программа управления внешним устройством компьютера.", "Набор микропрограмм, реализующих API для работы с аппаратурой компьютера и подключёнными к нему устройствами.", "Основной накопителель данных в большинстве компьютеров.", "Электронные и механические части вычислительного устройства."},

new string[]{ "5. Часть пользовательского интерфейса Microsoft Windows, используемая для настройки системы (панель управления)"},

new string[]{ "6. Как часто вы платите за интернет?", "1 раз в месяц.", "Я не плачу за него.", "1 раз в 6-12 месяцев.", "1 раз в несколько лет." },

new string[]{ "7. Сколькими устройствами вы пользуетесь?"},

new string[]{ "8. Есть ли у вас персональный компьютер?", "Да.", "Нет."},

new string[]{ "9. Как часто вы пользуетесь компьютером?", "Никогда", "почти никогда", "редко", "довольно-таки часто" },

new string[]{ "10. Вам пришло сообщение о выигрыше крупной суммы от неизвестного пользователя! Ваши действия:", "Сообщить всем своим родственникам", "Отослать все необходимые данные", "IT'S A TRAP!", "Не стану ничего высылать, проигнорировав сообщение" },

new string[]{ "11. Есть ли у вас антивирус?", "Да!", "Нет."},

new string[]{ "12. Часто ли вы скачиваете файлы с незнакомых сайтов?", "Каждый раз, когда мне необходимы какие-либо файлы", "Я пользуюсь только официальными источниками", "Довольно редко скачиваю", "Стараюсь не скачивать файлы с незнакомых сайтов" },

new string[]{ "13. Сопоставьте различные виды киберугроз с их определениями:", "Засорение электронной почты множеством писем", "Вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей", "Взлом ПК из любой точки мира", "Специальным образом организованные массированные сетевые запросы" },

new string[]{ "14. Что делать если у вас взломали аккаунт?", "Обратиться в техподдержку, одновременно с этим меняя пароли на связанных с этим аккаунтами.", "Выполнить все условия мошенников.", "Игнорировать взлом.", "Удалить свой аккаунт." },

};

}

}

Класс «Uchastnik»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Курсовая\_работа\_ИЦГН\_

{

public class Uchastnik

{

private string name;

public string Name

{

get

{

return name;

}

set

{

name = value;

}

}

private int age;

public int Age

{

get

{

return age;

}

set

{

age = value;

}

}

private string sex;

public string Sex

{

get

{

return sex;

}

set

{

sex = value;

}

}

public Uchastnik(string name, int age, string sex)

{

Name = name;

Age = age;

Sex = sex;

}

public Uchastnik()

{

}

}

}

Класс «Otver»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Курсовая\_работа\_ИЦГН\_

{

class Otvet:Uchastnik

{

private string[] cotveti = new string[14];

private string[] cot1 = new string[4];

private string[] cot2 = new string[4];

public string[] Cot1

{

get

{

return cot1;

}

set

{

cot1 = value;

}

}

public string[] Cot2

{

get

{

return cot2;

}

set

{

cot2 = value;

}

}

public string[] Cotveti

{

get

{

return cotveti;

}

set

{

cotveti = value;

}

}

public Otvet(string[] cotveti,string[] ot1,string[] ot2)

{

Cotveti = cotveti;

Cot1 = ot1;

Cot2 = ot2;

}

public Otvet()

{ }

public double razd(double timer)

{

double ktime = 0;

if(timer > 600)

{

ktime = 1;

}

return ktime;

}

}

}

Класс «CK»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Курсовая\_работа\_ИЦГН\_

{

class CK:Otvet

{

public CK()

{ }

public double KCK(string[] cotveti, string[] cot1)

{

double k1 = 0;

if (cotveti[0] == "Электронный блок либо интегральная схема, исполняющая машинные инструкции.")

{

k1 += 0.2;

}

if (cotveti[1] == "Чищу внутренние и внешние части компьютера с помощью специальных чистящих средств.")

{

k1 += 0.2;

}

if (cot1[0] == "Из материнской платы, сетевой карты, ЦП, Видеокарты и т.д.")

{

k1 += 0.1;

}

if (cot1[2] == "Устройств ввода и вывода, системного блока.")

{

k1 += 0.1;

}

if (cotveti[3] == "Набор микропрограмм, реализующих API для работы с аппаратурой компьютера и подключёнными к нему устройствами.")

{

k1 += 0.2;

}

if (cotveti[4] == "панель управления")

{

k1 += 0.2;

}

return k1;

}

public double KCP(string[] cotveti)

{

double k2 = 0;

if (cotveti[5] == "1 раз в месяц.")

{

k2 += 0.25;

}

if (cotveti[6] == "3" || cotveti[6] == "4" || cotveti[6] == "5" || cotveti[6] == "6" || cotveti[6] == "7" || cotveti[6] == "8" || cotveti[6] == "9" || cotveti[6] == "10")

{

k2 += 0.25;

}

if (cotveti[7] == "Да.")

{

k2 += 0.25;

}

if (cotveti[8] == "довольно-таки часто")

{

k2 += 0.25;

}

return k2;

}

public double KCB(string[] cotveti, string[] cot2)

{

double k3 = 0;

if (cotveti[9] == "Не стану ничего высылать, проигнорировав сообщение" || cotveti[9] == "IT'S A TRAP!")

{

k3 += 0.2;

}

if (cotveti[10] == "Да!")

{

k3 += 0.2;

}

if (cotveti[11] == "Я пользуюсь только официальными источниками")

{

k3 += 0.2;

}

if (cot2[0] == "Спам")

{

k3 += 0.05;

}

if (cot2[1] == "Фишинг")

{

k3 += 0.05;

}

if (cot2[2] == "Удалённый взлом")

{

k3 += 0.05;

}

if (cot2[3] == "Ddos-атаки")

{

k3 += 0.05;

}

if (cotveti[13] == "Обратиться в техподдержку, одновременно с этим меняя пароли на связанных с этим аккаунтами.")

{

k3 += 0.2;

}

return k3;

}

}

}

Класс «CK2»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Курсовая\_работа\_ИЦГН\_

{

class CK2: Otvet

{

public CK2()

{ }

public Tuple <double,double ,double ,double ,double> KCK2(string[] cotveti, string[] cot1)

{

double t1 = 0;

double t2 = 0;

double t3 = 0;

double t4 = 0;

double t5 = 0;

if (cotveti[0] == "Электронный блок либо интегральная схема, исполняющая машинные инструкции.")

{

t1 = 1;

}

if (cotveti[1] == "Чищу внутренние и внешние части компьютера с помощью специальных чистящих средств.")

{

t2 = 1;

}

if (cot1[0] == "Из материнской платы, сетевой карты, ЦП, Видеокарты и т.д.")

{

t3 += 0.5;

}

if (cot1[2] == "Устройств ввода и вывода, системного блока.")

{

t3 += 0.5;

}

if (cotveti[3] == "Набор микропрограмм, реализующих API для работы с аппаратурой компьютера и подключёнными к нему устройствами.")

{

t4 = 1;

}

if (cotveti[4] == "панель управления")

{

t5 = 1;

}

return new Tuple<double, double, double, double, double>(t1,t2,t3,t4,t5);

}

public Tuple<double, double, double, double> KCP2(string[] cotveti)

{

double t6 = 0;

double t7 = 0;

double t8 = 0;

double t9 = 0;

if (cotveti[5] == "1 раз в месяц.")

{

t6 = 1;

}

if (cotveti[6] == "3" || cotveti[6] == "4" || cotveti[6] == "5" || cotveti[6] == "6" || cotveti[6] == "7" || cotveti[6] == "8" || cotveti[6] == "9" || cotveti[6] == "10")

{

t7 = 1;

}

if (cotveti[7] == "Да.")

{

t8 = 1;

}

if (cotveti[8] == "довольно-таки часто")

{

t9 = 1;

}

return new Tuple<double, double, double, double>(t6, t7, t8, t9);

}

public Tuple<double, double, double, double, double> KCB2(string[] cotveti, string[] cot2)

{

double t10 = 0;

double t11 = 0;

double t12 = 0;

double t13 = 0;

double t14 = 0;

if (cotveti[9] == "Не стану ничего высылать, проигнорировав сообщение" || cotveti[9] == "IT'S A TRAP!")

{

t10 = 1;

}

if (cotveti[10] == "Да!")

{

t11 = 1;

}

if (cotveti[11] == "Я пользуюсь только официальными источниками")

{

t12 = 1;

}

if (cot2[0] == "Спам")

{

t13 += 0.25;

}

if (cot2[1] == "Фишинг")

{

t13 += 0.25;

}

if (cot2[2] == "Удалённый взлом")

{

t13 += 0.25;

}

if (cot2[3] == "Ddos-атаки")

{

t13 += 0.25;

}

if (cotveti[13] == "Обратиться в техподдержку, одновременно с этим меняя пароли на связанных с этим аккаунтами.")

{

t14 += 1;

}

return new Tuple<double, double, double, double,double>(t10, t11, t12, t13,t14);

}

}

}

Класс «Special»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Курсовая\_работа\_ИЦГН\_

{

class Special

{

public static string[] speotveti = new string[14];

public static string[] specot1 = new string[4];

public static string[] specot2 = new string[4];

public static List<Uchastnik> specPeople = new List<Uchastnik>();

public static double specicgn = 0;

}

}

Класс «ICGN»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Курсовая\_работа\_ИЦГН\_

{

class ICGN:CK

{

private string[] veti;

public string[] Veti

{

get

{

return veti;

}

set

{

veti = value;

}

}

private string[] ot1;

public string[] Ot1

{

get

{

return ot1;

}

set

{

ot1 = value;

}

}

private string[] ot2;

public string[] Ot2

{

get

{

return ot2;

}

set

{

ot2 = value;

}

}

private double t;

public double T

{

get

{

return t;

}

set

{

t = value;

}

}

public ICGN(string[] cotveti,string[] cot1, string[] cot2, double timer)

{

Veti = cotveti;

Ot1 = cot1;

Ot2 = cot2;

T = timer;

}

public double itograch()

{

double ktime = razd(t);

double k1 = KCK(veti, ot1);

double k2 = KCP(veti);

double k3 = KCB(veti, ot2);

double icgn = 0;

icgn = ((k1 + k2 + k3) / 3) \* 100 - 5 \* ktime;

return icgn;

}

}

}

Класс «Excell»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Microsoft.Office.Interop.Excel;

using Excel = Microsoft.Office.Interop.Excel;

namespace Курсовая\_работа\_ИЦГН\_

{

class Excell

{

public string path; //путь

\_Application excel = new Excel.Application(); //создаём наше приложение

Microsoft.Office.Interop.Excel.Workbook wb;

Microsoft.Office.Interop.Excel.Worksheet ws;

//Workbook wb; //книга

//Worksheet ws; //страница

public Worksheet ws1;

public Worksheet ws2;

public Worksheet ws3;

public Excell(string path, int sheet)// открываем файл на нужной странице

{

this.path = path;

wb = excel.Workbooks.Open(path);

ws = wb.Worksheets[sheet];

}

public string ReadCell(int i, int j)// читаем значение/текст из ячейки

{

if (ws.Cells[i, j].Value2 != null)

return ws.Cells[i, j].Value2.ToString();

else return "";

}

public void WritetoCell(int i,int j,string c)//записываем текст в ячейку

{

ws.Cells[i, j].Value2 = c;

}

public void Save()

{

wb.Save();

}

public void Close()

{

ws = null;

wb.Close(false, false, false);

excel.Quit();

wb = null;

excel = null;

GC.Collect();

}

public void SaveAs(string path)

{

wb.SaveAs(path);

}

}

}