**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Брянский филиал федерального государственного бюджетного образовательного учреждения высшего образования

**«Российский экономический университет имени Г.В. Плеханова»**

**Курсовая работа**

ПМ 01 «Разработка программных модулей программного обеспечения для

компьютерных систем»

МДК 01.02 «Прикладное программирование»

Специальность 09.02.03 «Программирование в компьютерных системах»

Тема: Разработка тестирующей программы по дисциплине «Информатика»

**Пояснительная записка**

Листов: 17

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_/И.Г.Стефашина/

«\_\_\_»\_\_\_\_\_\_\_\_\_2022 г.

Исполнитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Е.А.Иванова/

«\_\_\_»\_\_\_\_\_\_\_\_\_2022 г.

2022

Оглавление

[**Введение** 3](#_Toc119624107)

[**Глава 1. Теоретическая часть** 4](#_Toc119624108)

[**1.1 Методика использования контрольно-диагностических систем на уроках информатики** 4](#_Toc119624109)

[**1.2 Целесообразность компьютерного тестирования** 5](#_Toc119624110)

[**1.3 Анализ выбранной среды программирования** 6](#_Toc119624111)

[**Глава 2. Практическая часть** 8](#_Toc119624112)

[**2.1 Работа с внешним видом программного продукта** 8](#_Toc119624113)

[**2.2 Работа с кодом программного продукта** 8](#_Toc119624114)

[**Заключение** 11](#_Toc119624115)

[**Список использованных источников** 13](#_Toc119624116)

[**Приложение** 14](#_Toc119624117)

[**Листинг программы** 14](#_Toc119624118)

# **Введение**

Необходимо создать тестирующую программу, которая будет проверять знания тестируемого человека. Оглядываясь назад, в эпоху становления средств вычислительной техники, можно условно выделить ступени развития автоматизированных обучающих систем. Первые, самые ранние, представляли собой лишь электронные версии печатных изданий и содержали только текстовую информацию. Достоинством таких систем в сравнении с бумажными аналогами, по существу, являлась лишь интеграция информации. Постепенно, с развитием вычислительной техники, их стали сменять системы, которыми можно было пользоваться и как справочниками, то есть поисковые системы. Далее арсенал автоматизированных обучающих систем пополняется статической графикой, компонентами мультимедиа (звук и видео), компьютерной анимацией. Таким образом, системы развивались в области информационных баз знаний, поддерживающих уже не только текстовое, но и аудио - видео воздействие на обучаемого. Колоссальный скачок был сделан и в сфере развития систем контроля. Итак, средства контроля обеспечивают ввод ответа обучаемого, обработку полученной информации и выдачу управляющего воздействия (например, вы вводите ответ, а система одобряет или отвергает его).

Целью курсовой работы является разработка тестирующей программы по дисциплине “Информатика”. Приложение должно обеспечить ввод и регулирование вводимой информации.

Для этого были следующие задачи:

* Разработка интерфейса прикладной программы;
* Использование среды программирования Visual Studio;
* Разработка самой программы;

Предметом исследования являются возможность разработки прикладной программы на среде программирования Visual Studio.

Предполагается, что использование данной программы позволит проводить проверку учеников на знание дисциплины “Информатика”.

# **Глава 1. Теоретическая часть**

## **1.1 Методика использования контрольно-диагностических систем на уроках информатики**

Контроль уровня знаний и умений учащихся осуществляется путем проведения тестирования на разных этапах урока. Входное тестирование проводится перед началом работы с целью формирования индивидуальной учебной программы слушателя на основе выявления имеющихся умений и навыков соответствующего профессионального направления, которые он приобрел за период предварительной трудовой деятельности. Для осуществления входного тестирования преподаватель должен разработать перечень тестовых заданий, позволяющих выявить уровень знаний, умений и навыков, которыми владеет ученик по данной профессии. Для этого рекомендуется воспользоваться тестами к учебным элементам, модульным блокам и квалификационным тестам. Контроль достижений учащихся в процессе обучения, целью которого является выявление полноты овладения учащимся навыками, предусмотренными каждым учебным элементом, осуществляется путем тестирования.

Формой промежуточного контроля является тестирование, проводимое после окончания учащимся определенного этапа обучения, целью которого является выявление достаточных умений и навыков в выполнении отдельного вида работ по модульному блоку.

Такая методика применения тестовых систем обеспечивает:

* повышение заинтересованности учащихся в обучении, возрастает уровень знаний;
* увеличение объема самостоятельной работы учащихся в обучении;
* уменьшение нагрузки на преподавателя в подготовке и осуществлении контроля за уровнем знаний, умений и навыков учащихся;
* возможность преподавателю анализировать структуру знаний каждого учащегося и в соответствии с этим создавать и внедрять гибкую технологию обучения в зависимости от контингента учащихся;
* возможность осуществления индивидуального подхода к каждому учащемуся в зависимости от структуры его знаний с соответствующим выбором методики обучения;
* осуществление дифференциации учащихся по уровню знаний;
* объективность процесса оценки учебных достижений учащихся на уроках технологий.

Реализуя любую систему тестирования знаний учащихся, следует учитывать область использования тестов, количество учащихся, наличие человеческих и механических ресурсов для реализации тестов. Рассмотренная система использования всех видов тестирования знаний учащихся с помощью компьютера и компьютерной сети наиболее эффективна.

Наряду с положительным в тестировании есть определенные недостатки:

* составление тестовых заданий требует определенной квалификации и по силам далеко не каждому педагогу;
* отдельные ответы должны содержать элементы подсказок;
* тестирование позволяет достаточно надежно проверять знания, уровень сформированности умений, профессионального мышления, однако с помощью педагогических тестов можно проверять только опосредованно.

Обучение по модульной технологии с дискретным распределением материала на учебные элементы, каждый из которых имеет индикатор обратной связи, тест достижений, превращает постоянные контакты преподавателя с учащимися в неотъемлемую составляющую учебного процесса. Это позволяет получать исчерпывающую информацию о прогрессе в обучении каждого учащегося в каждой точке его учебной программы.

## **1.2 Целесообразность компьютерного тестирования**

Система компьютерного тестирования очень удобна для оценки знаний учащихся с низким и средним уровнем достижений. Она лишает учителя необходимости тратить драгоценное время урока на выслушивание нелепых устных ответов тех учеников, которые по определенным причинам не желают иметь больших успехов по предмету. В то же время она провоцирует их к выходу на определенный достаточный уровень знаний.

В отличие от учителя компьютерная программа оценивает не личность учащегося, а уровень его компетентности по данному предмету. Интересно мнение учащихся, преимущественно ребят, выбирающих предметы, ориентированные на технику: «Это очень честная система контроля знаний. Здесь важно не то, как сказал, а что ты знаешь».

## **1.3 Анализ выбранной среды программирования**

В данной курсовой работе будем рассматривать среду программирования от компании Microsoft под названием Visual Studio 2019 Community.

Visual Studio является интегрированной средой разработки (IDE) от компании Microsoft. С помощью данной среды можно разрабатывать:

* Мобильные приложения для операционных систем Windows, iOS и Android;
* Базы данных SQL Azure и SQL Server;
* Классические приложения для компьютера под управлением операционной системы Windows;
* Web-приложения;
* Различные расширения для Office, SharePoint, а также создание собственных расширений для Visual Studio;
* Облачные приложения;
* Игры;

В рассматриваемой среде есть возможность использовать такие технологии и языки программирования, как JavaScript, Python, Node.js, .NET, C#, F#, C, Visual Basic, C++.

Среда Visual Studio 2019 выпускается в трёх редакциях:

* **Community** – это бесплатная версия сред. Чтобы ее использовать, достаточно создать учетную запись Visual Studio, иначе она будет действовать только 30 дней. Имеет меньший функционал в сравнении с платными редакциями, но включает все необходимое для создания полноценных приложений. Походит для обучения и индивидуальных разработчиков;
* **Professional** – данная редакция содержит профессиональные инструменты для разработки приложений, однако функционал все равно не полный. Приходится для небольших групп разработчиков;
* **Enterprise** – это полнофункциональная версия Visual Studio 2019. Подходит для групп любого размера с высокими требованиями к качеству и масштабу.

Среду Visual Studio 2019 можно установить и работать в ней на следующих операционных системах:

* Windows 7 с Service Pack 1;
* Windows 8.1 (с обновлением 2919355);
* Windows 10 (1703 и выше);
* Windows Server 2012 R2 (с обновлением 2919355);
* Windows Server 2016 (Standard и Datacenter);
* Windows Server 2019 (Standard и Datacenter).

**Минимальные требования к оборудованию:**

* Процессор с тактовой частотой не ниже 1,8 ГГц. Рекомендуется использовать как минимум двухъядерный процессор;
* 2 ГБ оперативной памяти, рекомендуется 8 ГБ (если устанавливать на виртуальную машину, то минимум 2.5 ГБ);
* Свободного места на жестком диске от 800 мегабайт **до 210 гигабайт**, в зависимости от установленных компонентов. В большинстве случаев выделяйте как минимум 30 гигабайт, например, я устанавливал всего несколько компонентов, и у меня заняло более 20 ГБ места на жестком диске. Также Microsoft рекомендует устанавливать Visual Studio на SSD диск, я как раз устанавливал на SSD (на своем ноутбуке), скорость работы Visual Studio меня, конечно, порадовала;
* Видеоадаптер с минимальным разрешением 1280 на 720 пикселей (для оптимальной работы Visual Studio рекомендуется разрешение 1366 на 768 пикселей и более высокое).

**Дополнительные важные моменты:**

* Для установки Visual Studio 2019 требуются права администратора;
* Для работы Visual Studio 2019 требуется платформа .NET Framework 4.7.2, она будет установлена во время установки среды;
* Варианты «Основные серверные компоненты» и «Минимальный серверный интерфейс» не поддерживаются при запуске на Windows Server;
* Запуск Visual Studio 2019 (Professional, Community и Enterprise) в контейнерах Windows не поддерживается;
* Для интеграции Team Foundation Server 2019 с Office требуется Office 2016, Office 2013 или Office 2010;
* Для установки компоненты «Разработка мобильных приложений на C++, JavaScript или .NET» в ОС Windows 7 требуется PowerShell 3.0 или более поздняя версия;
* Visual Studio Community 2019 требуется активировать в течение 30 дней после установки, для этого необходимо создать учетную запись.

# **Глава 2. Практическая часть**

## **2.1 Работа с внешним видом программного продукта**

В данной курсовой работе предложено разработать тестирующую программу по дисциплине “Литература”. Она должна будет производить тесты и выдавать полученную оценку пользователю по результату тестирования. Для разработки данной прикладной программы будет использоваться среда разработки Visual Studio 2019 Community (C# Windows Forms).

Для написания теста на С# будет использоваться windows формы для того, чтобы была хоть какая-то графика. Для создания проекта выбирается Windows.Forms. Для начала задаётся графика, то есть, из панели элементов, выставляется на форму компоненты: Label (текстовое обозначение, 1 штука), RadioButton (элемент переключателя и выбора ответа, 3 штуки), button (кнопка, 2 штуки). С помощью свойства компонента, выставляется размер и стиль в разделе Font, для компонента Label и RadioButton; для Button настраивается размер кнопки и параметры текста. После настройки всей графики на форме, можно перейти к написанию кода.

## **2.2 Работа с кодом программного продукта**

Для начала нажимается два раза, по каждому компоненту и самой форме, чтобы правильно сформировать функционал объекта кода. Для начала объявляются переменные типа данных int: question\_count (счётчик вопросов), correct\_answer (количество правильных ответов), wrong\_answers (количество неправильных ответов), correct\_answers\_number (номер правильного ответа), selected\_response (номер выбранного ответа). Далее нужно создать массив (array) со строковым типом данных – string[] для запоминания и хранения различной информации. Создаётся функция void Start() – для функции начала, в ней же начинается писаться код для чтения из файла: с помощью переменной encoding которой присваивается код (System.Text.Encoding.GetEncoding(65001)) для распознавания кириллицы, чтобы всё работало без ошибок; конструкцию try{} catch{} для определения всплывающих ошибок в текстовом файле. В форме try {} определяется директория, расположения файла, System.IO.Directory.GetCurrentDirectory()+”@\t. С расширением txt”, и переменной encoding), создаётся функция для чтения текста – this.Text, которая задаётся в текстовом файле с помощью Read.ReadLine(). Обнуляются переменные queсtion\_count и correct и wrong \_answers, инициализируется массив array, где присваивается ему количество элементов -6, с помощью функции new String[10] c запасом. В форму catch(Exception){} записывается функция MessageBox.Show(“error 1”), которая будет показывать ошибки, которые произошли. В функцию Start записывается функция для считывания вопроса – вопрос(). Создаётся текстовый файл – System.IO и добавляется строка - .StreamReader, с помощью которой будет считываться файл, и создаётся переменная Read, при помощи которой будет считываться информация из файла. Начинается задаваться текст на кнопки, через код формы в функции Form1\_Load: на одной кнопке “Следующий вопрос” и на второй “выйти”. Создаётся состояние переключение для всех RadioButton, куда подключается функция для управления событий - .CheckedChanged+= new EventHandler (состояние переключения). Вызывается функция Start(), которая будет выполняться после всех переключений RadioButton. В самой функции Form1 инициализируется функция вопрос() – void вопрос (){}. В функции будет происходит смена вопроса с помощью кода “Label1.Text=Read.ReadLine()”, считывается текст с файла на Label1. Текст ответов будет содержаться в radioButton1,2,3, поэтому в них и должен считываться текст. Каждой кнопке со свойством .Text присваивается функция Read.ReadLine(). Присваивается переменной “номер правильного ответа” правильный ответ - correct\_answer =int.Parse(Read.ReadLine()).Нужно перевести все переключатели в состояние “выключено” – RadioButton1.Checked=false; для каждой кнопки. Для правильного функционала, в первом вопросе, первая кнопка делается не активной, так как ответ будет не выбран, button1.Enabled = false;. Увеличивается счётчик вопросов на 1. Проверяется конец файла или нет – if(Read.EndOfStream==true) button1.Text = “завершить”;. Реализуется функция “состояние переключения”, после функции вопрос () записывается инициализация функции void состояние переключения (object sender, EvenArgs e){}. В кнопку Button1 записывается инициализация .Enabled=true, что делает её активной и ей можно воспользоваться, этой же кнопке присваивается корректное переключение, с помощью функции .Focus();. Создаётся локальная переменная p, где будет делаться проверка переключения – RadioButton p= (RadioButton)sender;. Создаётся имя переключателя, с помощью ещё одной переменной var t, которой присваивается проверка переключения .Name;. Записывается в эту функцию переменная номер выбранного ответа, которой присваивается int.Parse(t.Substring(11)) – где идёт обработка её, как строки. Сoздаётся файл, для этого открывается папка в проводнике, создаётся текстовый файл t.txt, где будут храниться вопросы, номер вопросов, в формате 1/6, 2/6 и т.д., ответы и номер правильного ответа. Создаётся такой же файл, только в папке bin>Debug>t.txt, иначе он может не считаться и работать не корректно. Переходится к функционалу возможности узнать оценки и пройти тест заново. Нажимается на форме на кнопку Button1 два раза и попадается в функцию void button1\_Click(){}. Вводится подсчёт правильных ответов if(selected\_response==correct\_answers\_number) получается, что это правильный ответ и можно увеличить количество правильных ответов на один, в противном случае, ведётся счёт неправильных ответов и увеличивается счёт неправильных ответов на один и запоминаются вопросы, на которые ответили неправильно, чтобы потом вывести их пользователю в конце теста. Берётся массив и записываются в него номера неправильного вопроса и присваивается Label1.Text. Реализуется возможность повторного запуска теста в той же функции, if(button1.Text==”начать тестирование заново”){}, в этой функции пишется код для выполнения теста заново. Переключатели делаются снова доступными для выбора: RadioButton1.Visible=true;, для каждой кнопки. Переходится к началу теста функцией Start(); и оператор return;. Проверяется, является ли кнопка тексту “завершить”, если да, то завершается тест и добавляется функция закрытия Read.Close(), делаются кнопки неактивными и присваивается значение .Visible=false;. Выводится средняя оценка за тест – Label1.Text=string.format(“Тестирование завершено .\n” + “Правильных ответов: {0} из {1}.\n ” + “Набранные баллы {2:F2}.”, correct\_answers, question\_count, (correct\_answers\* 5.0F) / quection\_count);

Делается вывод неправильных вопросов, записывается новая переменная var Str, которая равняется “Cписок ошибок ”+ “:\n\n”, если строка равняется “Список ошибок”, то перебирается массив из неправильных ответов и выводится. После того, как вывелся список ошибок, то если есть неправильные ответы, то выводится через MessageBox список наших вопросов, проверяется, не равно ли количество не правильных ответов 0 и через MessageBox.Show(Str,”тестирование завершено”); выводится текст. Если текст кнопки 1 равен “следующий вопрос”, то вызывается функция “вопрос”. Начинается записываться функция выхода в кнопке2: this.Close();. Проверяется программа на работоспособность. Программа работает корректно, без ошибок и прочих нюансов.

# **Заключение**

Разработанные программные продукты предназначены для пользователей с небольшим опытом работы в области ПК. Программа реализована на C# в среде программирования Visual Studio 2019 Community на компьютере стандартной конфигурации с операционной системой Windows 10. Из многих языков C# является наиболее универсальным и простым в изучении. C# поддерживает все новейшие функции объектно-ориентированного программирования в мощных средах программирования, таких как Visual Studio 2019 Community. Продукт запускается стандартными методами приложений Windows без необходимости установки какого-либо дополнительного программного обеспечения.

Подводя итог по разработке и сопровождению программного обеспечения, можно сделать вывод, что поставленные цели и задачи были успешно выполнены.

При подготовке к этой курсовой работе я закрепил свои знания, полученные в классе и на практике преподавания, и улучшил свои навыки программирования на C#.

# **Список использованных источников**

1. Википедия [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>, свободный. (Дата обращения: 15.11.2022).
2. Nastoysam.ru [Электронный ресурс]. Режим доступа: <https://nastoysam.ru/wifi/rukovodstvo-dlya-nachinayushhih-o-tom-kak-nachat-rabotu-s-visual-studio>, свободный. (Дата обращения: 05.11.2021).
3. Microsoft [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/designers/walkthrough-windows-forms-designer?view=vs-2019>, свободный. (Дата обращения: 15.11.2022).
4. Microsoft [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019>, свободный. (Дата обращения: 15.11.2022).
5. Заметки IT специалиста [Электронный ресурс]. Режим доступа: <https://info-comp.ru/programmirovanie/739-install-visual-studio-2019-community.html>, свободный. (Дата обращения: 15.11.2022).
6. koi.tspu.ru [Электронный ресурс]. Режим доступа: <https://koi.tspu.ru/koi_books/samolyuk/lek11.htm>, свободный. (Дата обращения: 14.11.2022).
7. Microsoft [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2019>, свободный. (Дата обращения: 15.11.2022).
8. Microsoft [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/get-started/create-app-visual-studio?view=netdesktop-5.0>, свободный. (Дата обращения: 15.11.2022).
9. PoComp.ru [Электронный ресурс]. Режим доступа: <https://pocomp.ru/kak-otkryt-windows-form-v-visual-studio-2019/>, свободный. (Дата обращения: 14.11.2022).

# **Приложение**

## **Листинг программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Runtime.Remoting.Channels;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp3

{

public partial class Form1 : Form

{

int quection\_count;

int correct\_answers;

int wrong\_answers;

string[] array;

int correct\_answers\_number;

int selected\_response;

System.IO.StreamReader Read;

public Form1()

{

InitializeComponent();

}

void start()

{

var Encoding = System.Text.Encoding.GetEncoding(65001);

try

{

Read = new System.IO.StreamReader(

System.IO.Directory.GetCurrentDirectory() +

@"\t.txt", Encoding);

this.Text = Read.ReadLine();

quection\_count = 0;

correct\_answers = 0;

wrong\_answers = 0;

array = new String[10];

}

catch (Exception)

{

MessageBox.Show("ошибка 1");

}

вопрос();

}

void вопрос()

{

label1.Text = Read.ReadLine();

radioButton1.Text = Read.ReadLine();

radioButton2.Text = Read.ReadLine();

radioButton3.Text = Read.ReadLine();

correct\_answers\_number = int.Parse(Read.ReadLine());

radioButton1.Checked = false;

radioButton2.Checked = false;

radioButton3.Checked = false;

button1.Enabled = false;

quection\_count = quection\_count + 1;

if (Read.EndOfStream == true) button1.Text = "Завершить";

}

void состаяниеперключение(object sender, EventArgs e)

{

button1.Enabled = true; button1.Focus();

RadioButton Переключатель = (RadioButton)sender;

var tmp = Переключатель.Name;

selected\_response = int.Parse(tmp.Substring(11));

}

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

{

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

if (selected\_response == correct\_answers\_number) correct\_answers =

correct\_answers + 1;

if (selected\_response != correct\_answers\_number)

{

wrong\_answers = wrong\_answers + 1;

array[wrong\_answers] = label1.Text;

}

if (button1.Text == "Начать тестирование сначала")

{

button1.Text = "Следующий вопрос";

radioButton1.Visible = true;

radioButton2.Visible = true;

radioButton3.Visible = true;

start(); return;

}

if (button1.Text == "Завершить")

{

Read.Close();

radioButton1.Visible = false;

radioButton2.Visible = false;

radioButton3.Visible = false;

label1.Text = String.Format("Тестирование завершено.\n" +

"Правильных ответов: {0} из {1}.\n" +

"Набранные балы: {2:F2}.", correct\_answers,

quection\_count, (correct\_answers \* 5.0F) / quection\_count);

button1.Text = "Начать тестирование сначала";

var Str = "Список ошибок " +

":\n\n";

for (int i = 1; i <= wrong\_answers; i++)

Str = Str + array[i] + "\n";

if (wrong\_answers != 0) MessageBox.Show(

Str, "Тестирование завершено");

}

if (button1.Text == "Следующий вопрос") вопрос();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

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

{

button1.Text = "Следующий вопрос";

button2.Text = "Выход";

radioButton1.CheckedChanged += new EventHandler(состаяниеперключение);

radioButton2.CheckedChanged += new EventHandler(состаяниеперключение);

radioButton3.CheckedChanged += new EventHandler(состаяниеперключение);

start();

}

}

}