Федеральное государственное образовательное бюджетное учреждение высшего образования

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ

РОССИЙСКОЙ ФЕДЕРАЦИИ»

(Финансовый университет)

**Колледж информатики и программирования**

ПМ.03 Участие в интеграции программных модулей

Группа: 4ПКС-218

УТВЕРЖДАЮ

Председатель предметно-цикловой комиссии программирования и баз данных

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /А.И. Пестов/

« » декабря 2021г.

**ПРОЕКТ КУРСОВОЙ**

**На тему: «Проектирование и разработка конструктора тестов на языке программирования C#»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Руководитель курсового проекта

\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.В. Морозова/

Исполнитель курсового проекта

\_\_\_\_\_\_\_\_\_\_\_\_\_ /С.Р. Шахбабян/

Оценка за проект: \_\_\_\_\_\_\_\_\_\_\_\_

« » декабря 2021г.

Москва

2021

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc89873758)

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc89873759)

[1.1 Предпроектное исследование предметной области 4](#_Toc89873760)

[1.2 Сравнительный анализ программ-аналогов 4](#_Toc89873761)

[1.3 Постановка задачи 6](#_Toc89873762)

[1.4 Характеристика инструментальных средств разработки 7](#_Toc89873763)

[ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 8](#_Toc89873764)

[2.1 Анализ требований и определение спецификаций программного обеспечения 8](#_Toc89873765)

[2.2 Проектирование программного обеспечения 11](#_Toc89873766)

[2.3 Разработка программного обеспечения 13](#_Toc89873767)

[2.4 Руководство по использованию программы 21](#_Toc89873768)

[ЗАКЛЮЧЕНИЕ 28](#_Toc89873769)

[СПИСОК ЛИТЕРАТУРЫ 29](#_Toc89873770)

# ВВЕДЕНИЕ

В современном образовании одним из направлений улучшения контроля знаний и умений обучающихся стала тестовая технология. Как отмечается в методике преподавания, применение тестов – это одно из рациональных и эффективных дополнений к способам проверки знаний, умений и навыков обучающихся на различных ступенях обучения. Кроме этого, тестирование стало одним из средств индивидуализации в учебном процессе, потому что предоставляет право учитывать имеющиеся психологические особенности обучающихся, которые препятствуют их успешной учебной деятельности. Одновременно тестовый контроль обладает ряд существенных превосходств над другими вариантами имеющегося контроля. Во-первых, он дает возможность проверить солидный объем выученного материала небольшими порциями и, во-вторых, при помощи тестирования получается быстро диагностировать знания учебным материалом большего количества учащихся. При этом твердая процесс проверки знаний обучающихся практически исключает субъективизм. Систематичность в использовании тестового контроля, как правило, вырабатывает у обучающихся организованность и тяга к состязательности в усвоении программного материала. Всё это дает абсолютное основание для утверждения актуальности данной работы.

Таким образом, целью данного курсового проекта является создание приложения по тестирование обучающихся, включающего в себя ядро в виде базы данных, благодаря которому вопросы и ответы смогут хранится, или же добавляться в удобном для пользователя формате.

Задачами курсового проекта являются:

* определение требований к платформе;
* проектирование и реализация базы данных;
* проектирование и реализация приложения;
* Создание руководства пользователя.

Объектом исследования является настольное приложение по тестирование как форма контроля задач.

Предметом исследования являются технологии разработки Windows приложений.

В пользу актуальности могу сказать, что печатных тестов становится всё меньше и меньше в пользу электронных, в связи переходом большинства учебных заведений в дистанционный режим и цифровизации учебных заведений. Если раньше, чтобы проверить знания в учебных заведениях ученики, могут списать у друг друга тест, то сейчас на компьютерах им это будет гораздо сложнее, в связи более продуманной системой оценивания знаний.

Проект будет реализован в среде разработки Visual studio. Код приложения будет написан на языке программирования С#, а в качестве графического пользовательского интерфейса будет применяться WPF. Для хранения данных будет использоваться система управления базами данных (СУБД) Microsoft SQL Server.

# ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## Предпроектное исследование предметной области

На современном этапе образования в общеобразовательном учреждении контроль и оценка знаний учащихся все чаще и чаще обеспечивается при помощи разнообразных тестов. Тесты, как одна из форм контроля, в российском образовании появились сравнительно недавно, в отличие от зарубежных стран, где по этому вопросу имеет богатый опыт.

Интерес к тестированию в отечественном образовании объясняется тем, что помимо своей основной функции - контроля знаний учащихся, оно может выполнять и другие функции:

* служит средством диагностики трудностей учебного материала,
* является мерой определения качества обученности;
* дает возможность для прогнозирования успешности или неуспешности обучения.

С таким приложением вырастет скорость проверки знаний обучающихся, следовательно вырастет эффективность обучения.

Пользователями данного программного продукта могут быть как Учителя, так и ученики. Следовательно приложение должно быть интуитивно понятным и простым в использовании.

Таким образом можно сделать вывод, что приложение должно включать в себя следующие функции:

* Регистрация, пользователей
* Создание теста
* Создание теории к тесту
* Выбор теста
* Проверка теста

## Сравнительный анализ программ-аналогов

myQuiz

Преимущества- лёгок в эксплуатации

Минусы:

легко взломать и найти ответы

веб версия

платный

Moodle

Преимущества

Большое количество учебных элементов поддерживает дифференцированное обучение

Минусы

Платно

Нет русского языка

Нет возможности установить свои модули.

На основании таблицы 1 можно сделать выводы о привлекательности моего приложения. У программ-аналогов есть свои преимущества и недостатки. Мое решение скомбинирует все плюсы и уберет недостатки. Все необходимые функции будут реализованы, при этом продукт будет бесплатный, что является очень большим плюсом.

Таблица 1. Сравнительная таблица аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Функции | myQuiz | Moodle | Мой проект |
| Создание теста | + |  | + |
| Создание теории |  |  | + |
| Вывод результатов тестирования | + | + | + |
| Интуитивно понятный интерфейс | + |  | + |
| Стоимость | От 30000 в месяц | От 80000 в месяц | 0 |

## Постановка задачи

Входными данными данные тестируемого (ФИО, если он ученик), вопросы теста, ФИО учителя;

Выходными данными являются результаты теста.

Для взаимодействия с ПО используется пользовательский интерфейс, в котором будут реализованы функции, перечисленные выше. Интерфейс должен быть простым и понятным в использовании.

Функциональные требования:

* Регистрация учителя
* Регистрация ученика
* Создания теста
* Создание теории к тесту

Эксплуатационные требования:

* Интерфейс приложения будет графическим и интуитивно понятным, что гарантирует удобство для пользователя.

Системные требования:

* Платформа [OS]: Windows
* Разрядность: х86(32-bit) x64(64-bit)
* ЦП [CPU]: от 500 MHz
* Видеоадаптер [GPU]: 3D
* Видеопамять [VRAM]: 64 Mb
* Винчестер [HDD]: 3Gb
* Оперативная память [RAM]: 256 Mb
* Контроллер: Клавиатура, Мышь

## Характеристика инструментальных средств разработки

Данная программа была написана на объектно-ориентированном языке программирования C# с помощью MS Visual Studio 19 с множеством различных встроенных библиотек и функций, удобным для пользователя интерфейсом и наличием отладчика кода.

Программирование выполнялось на языке C#, тип проекта – WPF Приложение.

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

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

# ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

## Анализ требований и определение спецификаций программного обеспечения

Был проведен анализ требований и определение спецификаций программного обеспечения, и составлены UseCase, Функциональная и «Сущность-связь» диаграммы.

На рисунке 1 представлена UseCase диаграмма, на которое показаны все функции доступные разным пользователям.

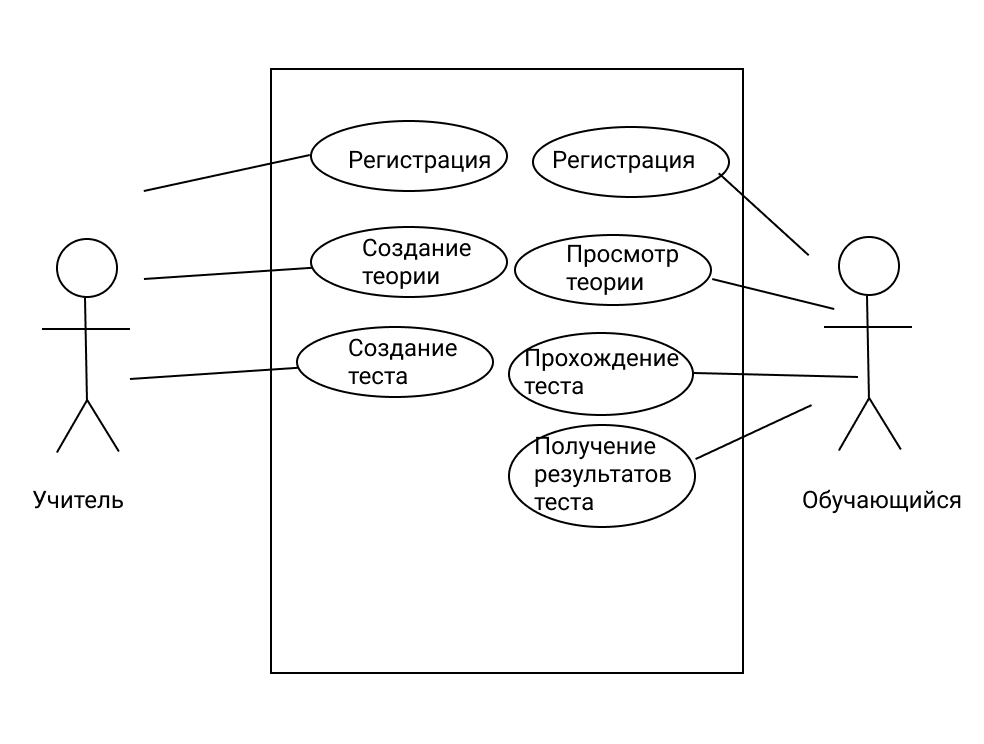


Рисунок 1. UseCase диаграмма

На рисунке 2 представлена функциональная диаграмма.

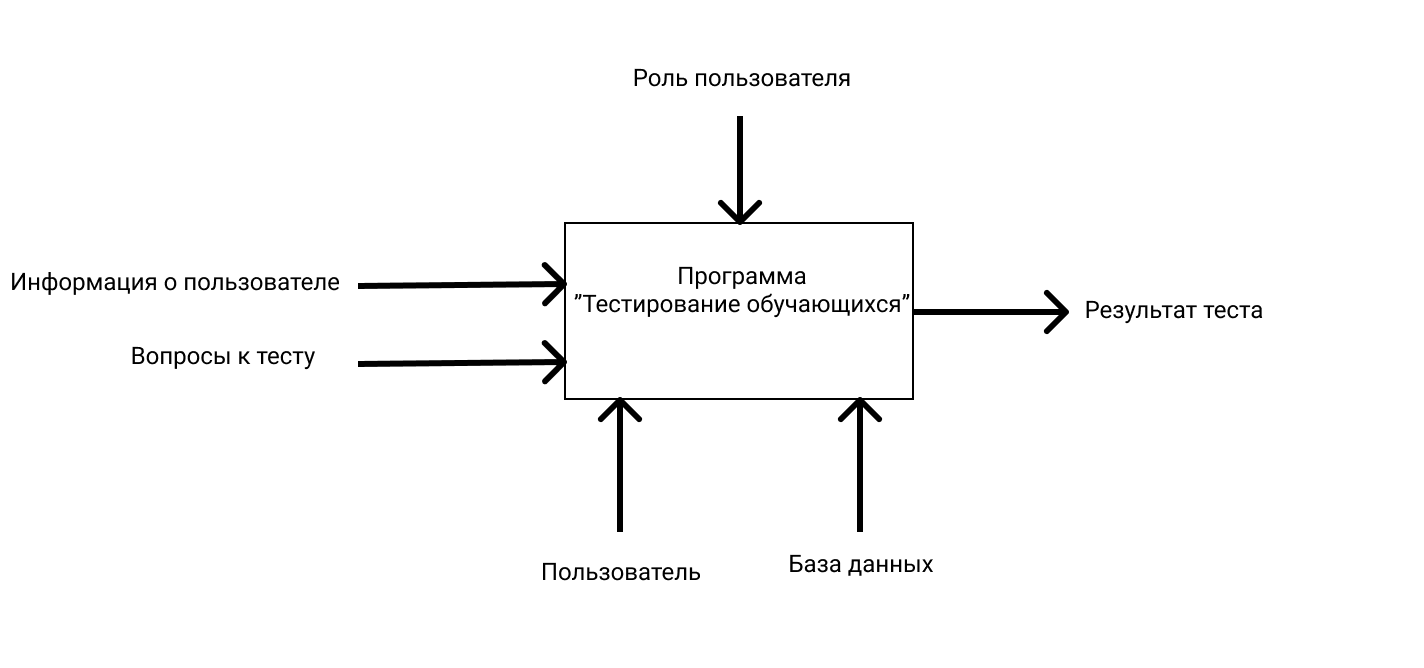


Рисунок 2. Функциональная диаграмма

На рисунке 3 представлена диаграмма «Сущность-связь»:

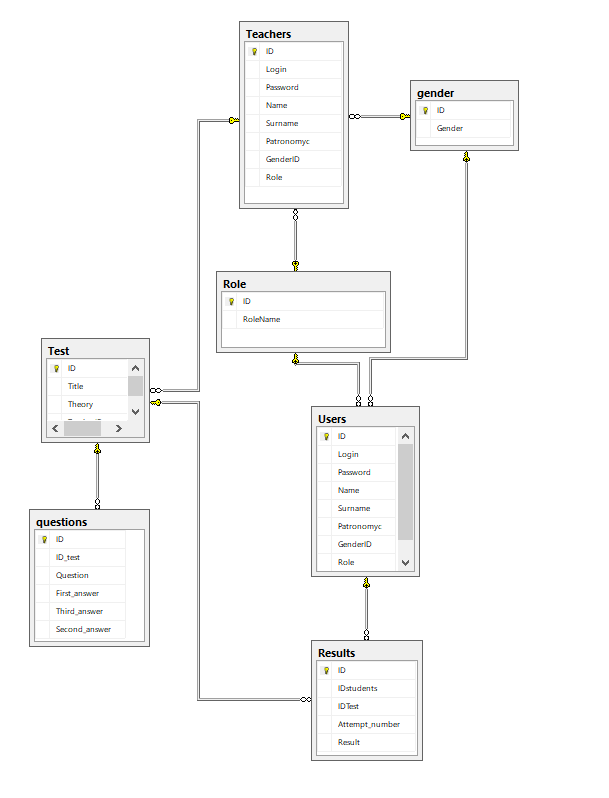


Рисунок 3. Диаграмма «Сущность-связь»

## Проектирование программного обеспечения

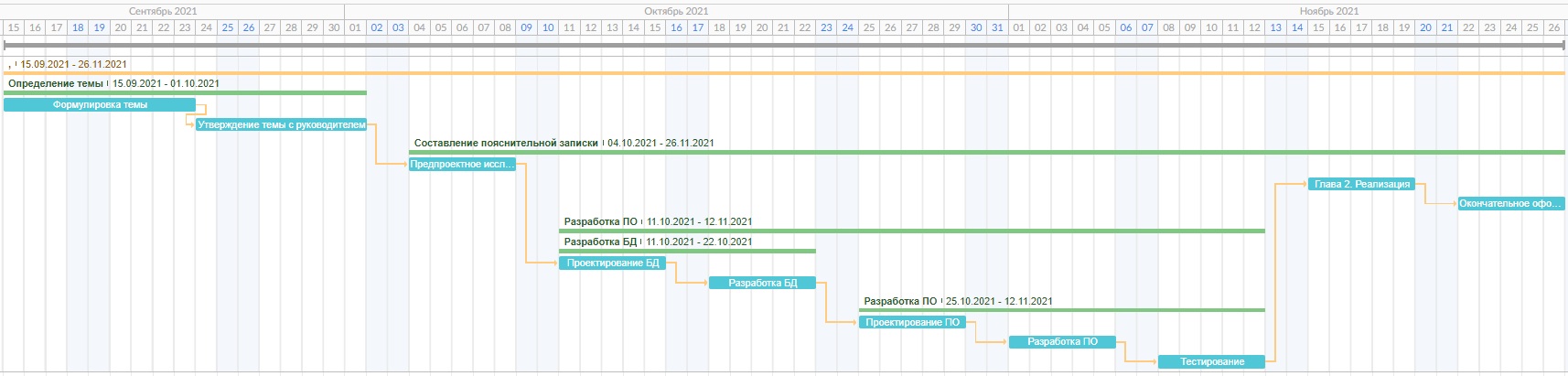
На первом этапе работы была составлена диаграмма Ганта, представленная на рисунке 4, где указаны сроки каждого этапа работы над курсовым проектом.

Рисунок 4. Диаграмма Ганта

После составления диаграммы Ганта была составлена структурная схема программы, представленная на рисунке 5:

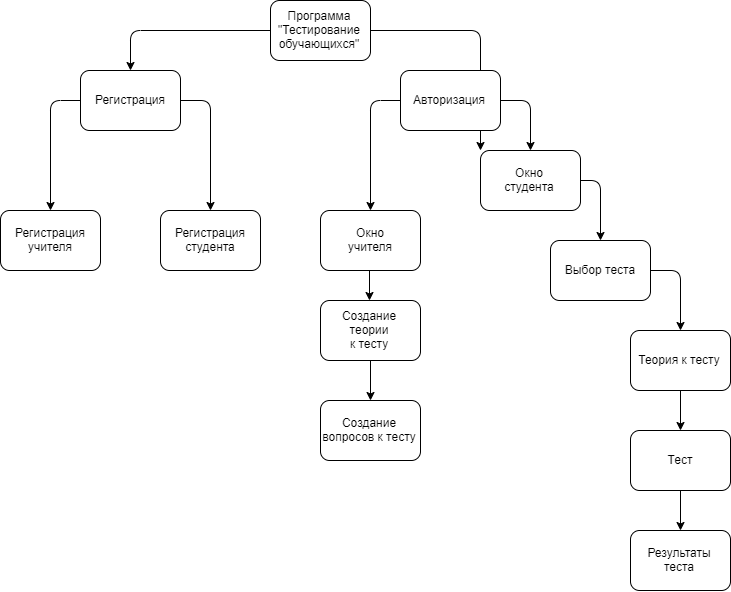


Рисунок 5. Структурная схема

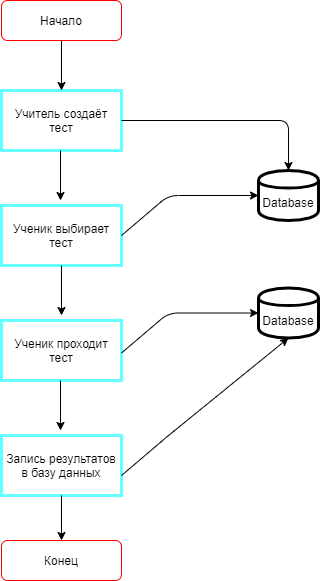
После составление структурной схемы, и определения функционала, была составлена функциональная схема, наглядно показывающая принцип работы программы. Функциональная схема представлена на рисунке 6:

Рисунок 6. Функциональная схема

## Разработка программного обеспечения

Приложение состоит из окон. Окно первое- вход в программу

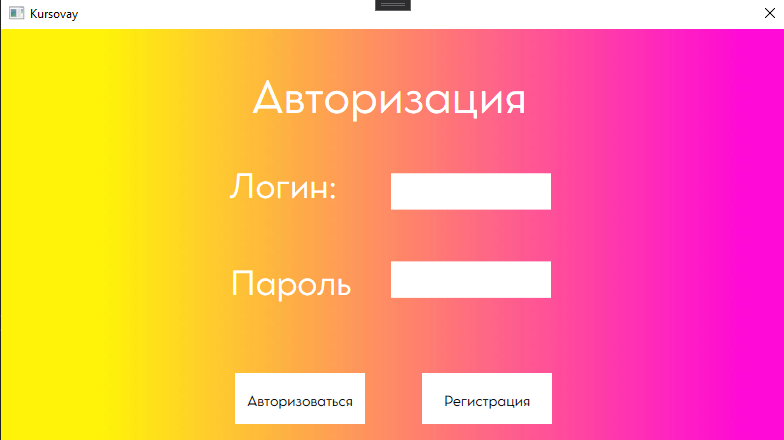


Рисунок 7. Страница “MainWindow”

Ниже представлен код входа в аккаунт:

private void Button\_autoriz(object sender, RoutedEventArgs e)

{

var users = Core.db.Users.ToList();

foreach (var p in users)

{

if (p.Password == pas.Password && p.Login == log.Text)

{

Mystudent = Core.db.Users.First(c => c.ID == p.ID);// сохраняем в лист информацию о том какой студент зашёл

test\_student ts = new test\_student(Mystudent);

ts.Show();

this.Close();

}

}

var teachers = Core.db.Teachers.ToList();// список всех учителей

foreach (var p in teachers)

{

if (p.Password == pas.Password && p.Login == log.Text)

{

MessageBox.Show("Пользователь авторизовался - учитель");

Teacher = Core.db.Teachers.First(c => c.ID == p.ID);// сохраняем в лист информацию о том какой учитель зашёл

Lesson L1 = new Lesson(Teacher);

L1.Show();

}

}}

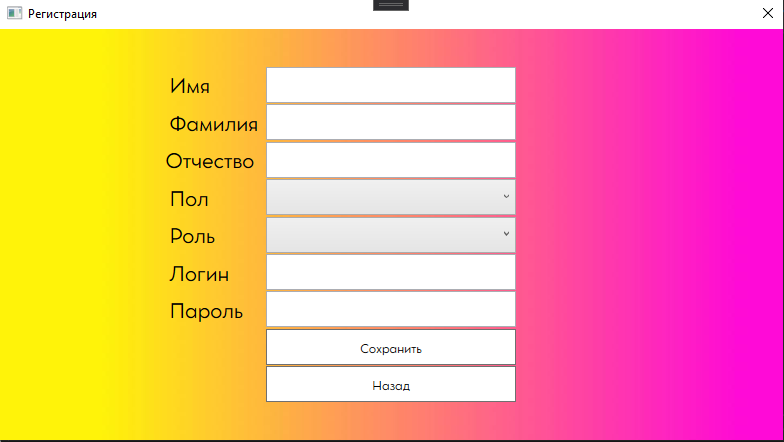


Рисунок 8. Страница “NewEmpPage”

На рисунке 8 продемонстрировано окно регистрации пользователя

В котором регистрирующий человек вводит данные о себе. Также выбирает роль – либо ученик либо учитель.

Ниже представлен код этого окна:

public partial class Reg : Window

{

public Reg()

{

InitializeComponent();

Role.ItemsSource = Core.db.Role.ToList();

GenderID.ItemsSource = Core.db.gender.ToList();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

if ((Role)Role.SelectedItem == Core.db.Role.First(c=>c.RoleName=="Учитель"))

{

Teachers teachers = new Teachers();

teachers.ID = Core.db.Teachers.ToList().Last().ID + 1;

teachers.Name = Name.Text;

teachers.Surname = Surname.Text;

teachers.Patronomyc = Patronymic.Text;

teachers.Login = Login.Text;

teachers.Password = Password.Password;

teachers.Role1 = (Role)Role.SelectedItem;

teachers.gender = (gender)GenderID.SelectedItem;

Core.db.Teachers.Add(teachers);

}

else

{

Users users = new Users();

users.ID = Core.db.Users.ToList().Last().ID + 1;

users.Name = Name.Text;

users.Surname = Surname.Text;

users.Patronomyc = Patronymic.Text;

users.Login = Login.Text;

users.Password = Password.Password;

users.Role1 = (Role)Role.SelectedItem;

users.gender = (gender)GenderID.SelectedItem;

Core.db.Users.Add(users);

}

Core.db.SaveChanges();

MainWindow main = new MainWindow();

main.Show();

this.Close();

}

private void GenderID\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void Back\_Click(object sender, RoutedEventArgs e)

{

MainWindow main = new MainWindow();

main.Show();

this.Close();

}

}

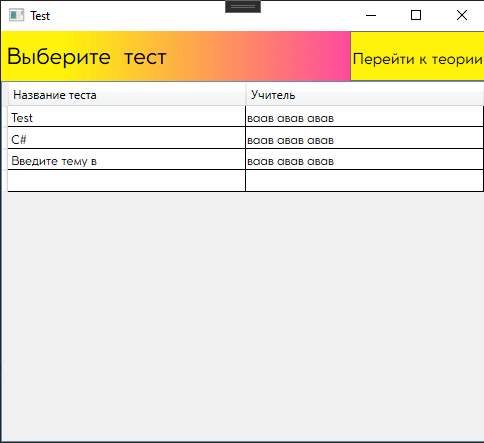


Рисунок 9. Окно выбора теста

Рисунок 9. В данном окне пользователь с ролью ученик выбирает из списка тестов, тест который необходимо пройти. И приступает к нему

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9. Окно теории к тесту

На окне под рисунком 9, обучающийся ознакамливаться с теорией перед началом теста. Кнопка перейти к тесту переводит обучающегося к самому тесту.

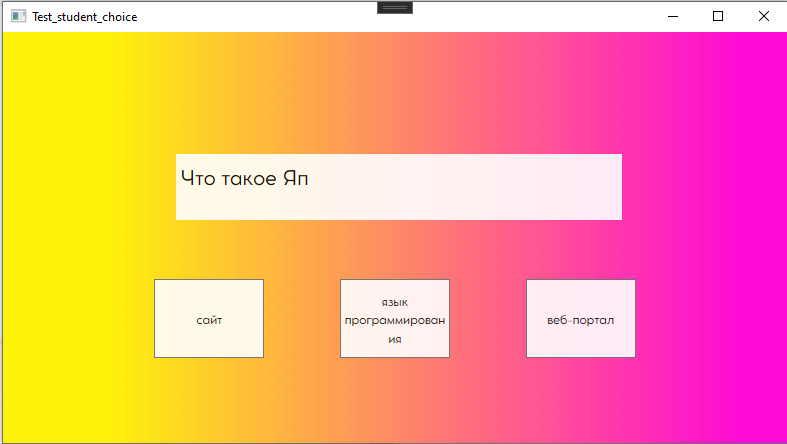


Рисунок 10. Окно теста

В данном окне студент проходит сам тест выбирая правильный вариант ответа.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11. Результат теста

В данном окне выводится результаты пройденного теста, ученика. Чем больше он ответит правильно, тем выше бал он получит

Ниже представлен код этого окна:

public partial class Result\_test : Window

{

public Users student { get; set; }

public Test Test { get; set; }

public Results Attempt\_number\_student { get; set; }

public Result\_test(int ans, int amount\_cor\_ans, Users users, Test test)

{

InitializeComponent();

DataContext = this;

var anss = ans;

var amount\_cor\_anss = amount\_cor\_ans;

Test = test;

student = users;

SSS.Text = student.Name + " " + student.Surname;

int bal = 0;

double b = (double)amount\_cor\_anss / anss;

double a = 0;

a = (double)b \* 100;

a = Math.Round(a, 2);

if (a >= 70 && a <= 90)

{

bal = 4;

Ball.Text = "4";

}

else

if (a >= 90 && a <= 100)

{

bal = 5;

Ball.Text = "5";

}

else

if (a >= 30 && a <= 70)

{

bal = 3;

Ball.Text = "3";

}

else

if (a >= 0 && a <= 30)

{

bal = 2;

Ball.Text = "2";

}

string bb = Convert.ToString(a);

Result.Text = bb + "%";

Results results = new Results();

results.ID = Core.db.Results.ToList().Last().ID + 1;

results.IDstudents = student.ID;

results.IDTest = Test.ID;

Attempt\_number\_student = Core.db.Results.First(c => c.IDstudents == student.ID);

results.Attempt\_number = Attempt\_number\_student.Attempt\_number + 1;

results.Result = bal;

Core.db.Results.Add(results);

Core.db.SaveChanges();

}

}

}

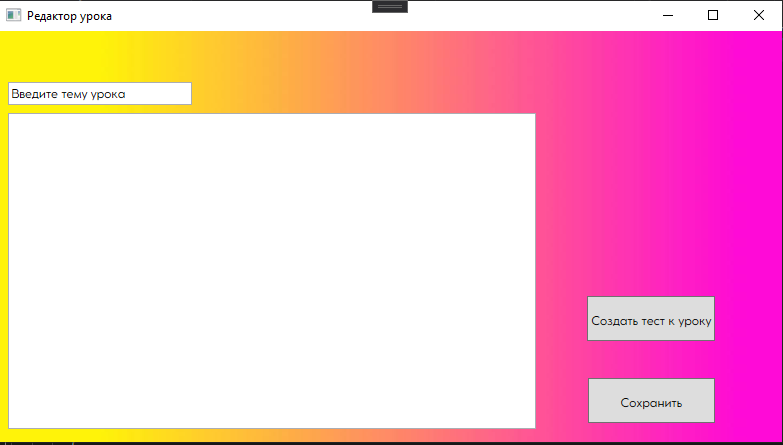


Рисунок 12. Создание теории к тесту

На рисунке 12, учитель создаёт теорию к тесту, которая будет показываться ученику, прежде чем тот начинает тест.

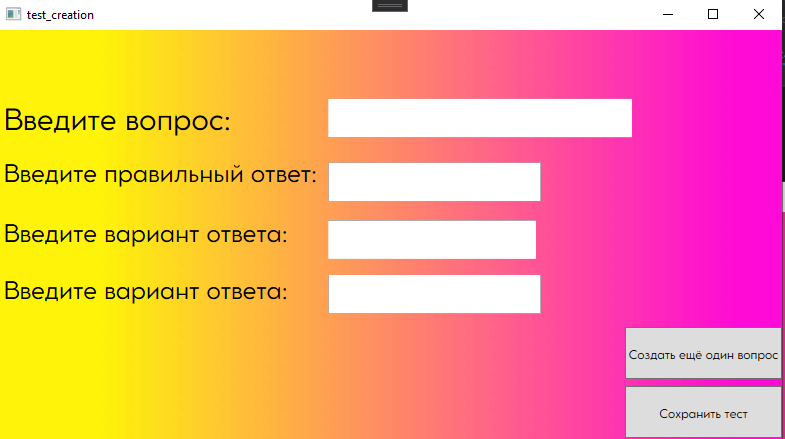


Рисунок 13. Окно создание вопросов к тесту

Рисунок 13, здесь учитель создаёт вопросы к тесту.

## Руководство по использованию программы

*Руководство системного программиста*

Общие сведения о программе

Программа предназначена для тестирования обучающихся. В программе доступны следующие функции:

* Регистрация пользователя
* Регистрация учителя
* Вход в аккаунт
* Создание теста для учеников
* Прохождение теста

Программный продукт разработан в среде программирования Visual Studio на языке программирования С#. В качестве графического пользовательского интерфейса выбран WPF. В качестве СУБД выбран Microsoft SQL Server.

Структура программы

На рисунке 14 представлена структурная схема:

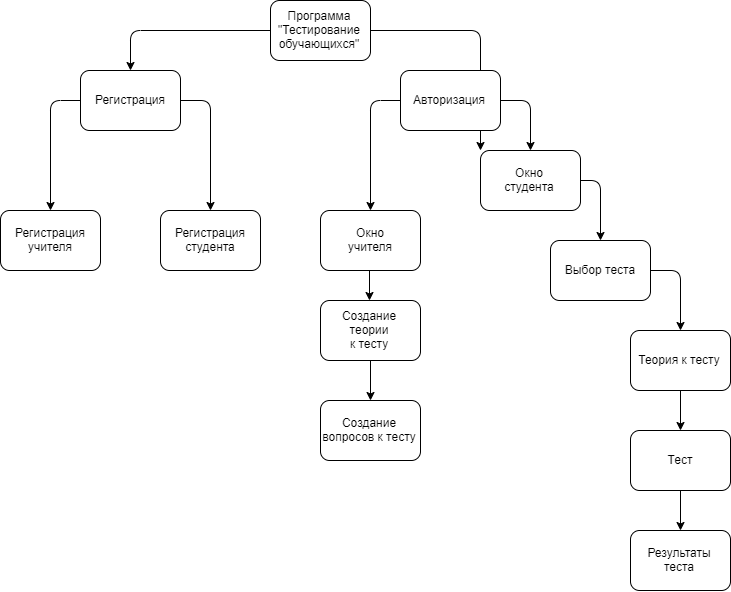


Рисунок 14. Структурная схема

Настройка программы

Для установки программы запустите файл setup.exe, после чего откроется окно, представленное на рисунке 15.

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 15. Окно принятия соглашения

После нажатия кнопки «Принимаю» произойдет загрузка .NET Framework нужной версии, и откроется окно, представление на рисунке 32.

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 16. Окно предупреждения о безопасности

После нажатия на кнопку «Установить» произойдет установка программы, и откроется сама программа.

После установки программу можно будет открыть через меню «Пуск» на вашем компьютере или через ярлык на рабочем столе. Удаление программы происходит через меню «Установка и удаление программ». При нажатии на кнопку удалить будет открыто окно, представленное на рисунке 33.

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 17. Окно удаления программы

После нажатия на кнопку «Ок» программа будет полностью удалена с компьютера.

Проверка программы

Программа имеет ряд способов проверки, такие как запрет ввода некорректных данных, генерация случайного кода компании в зависимости от выбора пользователя, появление всплывающих окон. Более подробно эти процессы описаны в параграфе 2.4 Отладка и тестирование программы.

Сообщения системному программисту

Сообщения системному программисту не предусмотрены.

*Руководство программиста*

Назначение и условия применения программы

Программа способна выполнить следующие функции:

* Регистрация пользователя
* Регистрация учителя
* Вход в аккаунт
* Создание теста для учеников
* Прохождение теста

Системные требования:

* Платформа [OS]: Windows
* Разрядность: х86(32-bit) x64(64-bit)
* ЦП [CPU]: от 500 MHz
* Видеоадаптер [GPU]: 3D
* Видеопамять [VRAM]: 64 Mb
* Винчестер [HDD]: 3Gb
* Оперативная память [RAM]: 256 Mb
* Контроллер: Клавиатура, Мышь

Обращение к программе

Для запуска программы необходимо запустить ее из меню «Пуск» или нажать на ярлык на рабочем столе.

Входные и выходные данные

Входными данными являются данные о пользователей, вопросы к тесту.

Выходными данными являются результаты пройденных тестов.

*Руководство оператора*

Назначение программы

Программа предназначена для тестирования обучающихся. В программе доступны следующие функции:

* Регистрация пользователя
* Регистрация учителя
* Вход в аккаунт
* Создание теста для учеников
* Прохождение теста

Условия выполнения программы

Системные требования:

* Платформа [OS]: Windows
* Разрядность: х86(32-bit) x64(64-bit)
* ЦП [CPU]: от 500 MHz
* Видеоадаптер [GPU]: 3D
* Видеопамять [VRAM]: 64 Mb
* Винчестер [HDD]: 3Gb
* Оперативная память [RAM]: 256 Mb
* Контроллер: Клавиатура, Мышь

Выполнение программы

Для запуска программы необходимо запустить ее из меню «Пуск» или нажать на ярлык на рабочем столе. Во время исполнения программы могут появится всплывающие окна. В них сказано какая допущена ошибка при вводе данных или будет выведена информация о успешном выполнении действия. Чтобы закрыть всплывающее окно нажмите на кнопку «Ок» или выберите вариант «Да» или «Нет», если такой выбор имеется.

Сообщения оператору

Во время исполнения программы могут появится всплывающие окна. В них сказано какая допущена ошибка при вводе данных или будет выведена информация о успешном выполнении действия. Чтобы закрыть всплывающее окно нажмите на кнопку «Ок» или выберите вариант «Да» или «Нет», если такой выбор имеется.

# ЗАКЛЮЧЕНИЕ

В итоге всей проделанной работы все цели, поставленные в предпроектном исследовании, были выполнены в ходе реализации программного продукта.

Все задачи курсового проектирования были выполнены. Были реализованны все функциональные требования:

* Регистрация пользователя
* Регистрация учителя
* Вход в аккаунт
* Создание теста для учеников
* Прохождение теста

Разработанный в рамках курсового проектирования программное средство имеет ряд достоинств:

* Программа интуитивно понятна;
* В программе сохранена эргономичность, что упрощает работу с программой для пользователя;

В результате самостоятельного изучения инструментальных средств и реализации данного проекта, мною были детально освоены навыки работы с базами данных, куда входит подключение базы к программе, изменение и удаление элементов, добавление новых элементов, привязка различных элементов управления к определенным столбцам или элементам в базе с помощью инструмента «Binding».

# СПИСОК ЛИТЕРАТУРЫ

*Законодательные и нормативные акты:*

1. ГОСТ Р 7.0.12-2011 Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2004. – 169 с.
2. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления
3. ГОСТ 7.32-2017 Отчет о научно-исследовательской работе. Структура и правила оформления
4. ГОСТ 7.82-2001 Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления
5. ГОСТ Р 7.0.100-2018 Библиографическая запись. Библиографическое описание. Общие требования и правила составления.
6. ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления
7. Единая система программной документации. – М.: Стандартинформ, 2005. – 128 с.

*Учебная и научная литература:*

11. Федорова, Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: Учебное пособие / Г.Н. Федорова. – М.: КУРС: ИНФРА-М, 2019. – 336 с. – (Среднее профессиональное образование).

*Интернет-документы*

13. Введение в язык C# и .NET Framework. – [Электронный ресурс]. – URL: <https://msdn.microsoft.com/ru-ru/library/z1zx9t92.aspx> (дата обращения: 25.09.2020)

14. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. – Москва: ФОРУМ: ИНФРА-М, 2020. – 400 с. – (Среднее профессиональное образование). – ISBN 978-5-8199-0812-9. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1067012> (дата обращения: 30.09.2020)

15. Гниденко, И. Г.  Технологии и методы программирования: учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Издательство Юрайт, 2020. – 235 с. – (Высшее образование). – ISBN 978-5-534-02816-4. – Текст : электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/450999> (дата обращения: 25.09.2020)

16. Гниденко, И. Г.  Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Издательство Юрайт, 2020. – 235 с. – (Профессиональное образование). – ISBN 978-5-534-05047-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/453640> (дата обращения: 26.09.2020)

17. Гуриков, С. Р. Введение в программирование на языке Visual C#: учебное пособие / С.Р. Гуриков. — МОСКВА: ФОРУМ: ИНФРА-М, 2020. — 447 с. — (Высшее образование: Бакалавриат). – ISBN 978-5-00091-458-8. ­– Текст: электронный. – URL: <https://znanium.com/catalog/product/1092167> (дата обращения: 29.09.2020)

18. Интернет-сервис для построения схем и диаграмм Draw.io. – [Электронный ресурс]. – URL: <https://www.draw.io/> (дата обращения: 30.09.2020)

19. Интернет-сервис для построения UML-диаграмм. – [Электронный ресурс]. – URL: <https://plantuml.com/> (дата обращения: 30.09.2020)

20. Казанский, А. А.  Программирование на Visual C#: учебное пособие для вузов / А. А. Казанский. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2020. – 192 с. – (Высшее образование). – ISBN 978-5-534-12338-8. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/451467> (дата обращения: 30.09.2020)

21. Полное руководство по языку программирования C# 6.0 и платформе .NET 4.6. – [Электронный ресурс]. – URL: <http://metanit.com/sharp/tutorial/> (дата обращения: 27.09.2020)

22. Руководство по программированию в Windows Forms. – [Электронный ресурс]. – URL: <http://metanit.com/sharp/windowsforms/> (дата обращения: 27.09.2020)

23. Руководство по программированию в WPF. – [Электронный ресурс]. – URL: <https://metanit.com/sharp/wpf/> (дата обращения: 29.09.2020)

24. Руководство по работе в среде Visual Studio. – [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/visualstudio/> (дата обращения: 25.09.2020)

25. Чистов, Д. В. Проектирование информационных систем: учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук; под общей редакцией Д. В. Чистова. – Москва: Издательство Юрайт, 2020. – 258 с. – (Профессиональное образование). – ISBN 978-5-534-03173-7. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/452680> (дата обращения: 26.09.2020)