СОДЕРЖАНИЕ

Введение 4

1 Технический проект 6

1.1 Анализ предметной области 6

1.2 Постановка задачи 6

1.3 Требования к программному обеспечению и техническим средствам 8

1.4 Проектирование функциональности и структуры программы 8

1.5 Разработка концептуальной модели базы данных 13

1.6 Выбор средств разработки 14

2 Рабочий проект 17

2.1 Проектирование физической модели базы данных 17

2.2 Функциональное взаимодействие модулей программы 17

2.3 Входные и выходные данные 20

2.4 Установка и настройка программы 20

2.5 Работа с программой 21

Заключение 25

Список использованных источников 27

Приложение А – Листинг модуля Question.cs 28

# ВВЕДЕНИЕ

Темой курсового проекта является разработка и создания приложения по теме «Космос».

Целью данного курсового проекта будет являться написание приложения, позволяющего быстро и познавательно рассмотреть основы космического пространства, а также проверить свои знания с помощью тестирования

Написание данного курсового проекта является целесообразным, потому что:

* приложение позволит усвоить учебный материал без использования дополнительной учебной литературы;
* приложение позволит проверить усвоение материала в интерактивной форме;
* на рынке программных изделий имеется мало подобных программ.

Для достижения поставленной цели будут рассмотрены и решены следующие задачи:

* разработана структура и функциональность программы;
* разработаны справочные и тестовые материалы;
* произведён выбор средств реализации;
* описаны минимальные требования к программно-техническим средствам;
* описаны входные и выходные данные;
* разработан программный код приложения;
* создано руководство пользователя;
* разработана инсталляция.

Объектом исследования будут являться космические тела и информационная среда, используемая для обучения и проверки знаний по теме «Космос».

Предметом исследования будут являться средства реализации справочного и контролирующего приложения.

Практическая значимость проекта заключается в возможности использования приложения в учебном процессе для самообучения и самоконтроля. Это значительно улучшит качество полученных знаний по теме.

Основными пользователями данной программы будут люди, желающие узнать больше о космосе.

Теоретический материал, а также тестовые задания будут браться из интернет-источников. Количество вариантов тестовых заданий будет достаточным для полного охвата всех вопросов по теме, а также закрепления полученных знаний.

Теоретическими основами и методами решения поставленных задач будут методологии проектирования функциональности приложения, структуры базы данных (БД), особенности реализации программного кода с помощью конкретного языка программирования, принципы и алгоритмы применения инструментальных средств на различных этапах разработки программного продукта.

# 1 ТЕХНИЧЕСКИЙ ПРОЕКТ

## 1.1 Анализ предметной области

С древних времен взоры людей были устремлены в небо. Начиная с первых шагов по земле, человек ощущал свою зависимость от неба, его жизнь и деятельность во многом зависели от него. Наши предки хорошо знали и разбирались в «повадках» неба. Для них небо было живым, наполненным, многообразно себя проявляющим.

Знание космоса в наше время стало актуальным так как, космические исследования являются мощным рычагом научно-технического прогресса. Они дают огромный теоретический и экспериментальный материал для развития наших знаний о Вселенной и Земле, происхождении планет и строении вещества, оказывают огромное влияние на развитие культуры и уже приносят практическую пользу в развитии космической связи, космической метеорологии и навигации. Успехи в решении сложнейших задач, связанных с исследованием и освоением космического пространства, вселяют уверенность в том, что и в дальнейшем на этом пути будут новые достижения, которые, в свою очередь, будут способствовать дальнейшему развитию и прогрессу человеческого общества [9].

Разрабатываемое приложение будет содержать материал по основным космическим объектам, а также тест по представленным данным.

## 1.2 Постановка задачи

Необходимо разработать обучающую и контролирующую программу по теме «Космос», а также создать документационное сопровождение программы, включающее в себе технический и рабочие проекты.

Материал по темам должен быть представлен в файлах, поддерживающих текст и изображения.

Данная программа должна будет содержать все необходимые сведения о космосе, иметь удобный и понятный пользователю графический интерфейс.

Программа должна обеспечивать выполнение следующих функций:

* отображение меню;
* переход между обучающей и контролирующей частями;
* возможность выбора темы изучения;
* отображение информации по выбранной теме;
* автоматизацию контроля знаний за счет автоматического проведения теста и его оценивания;
* выставление оценки с указанием количества набранных баллов;
* хранение материалов теста в БД.

Теоретический материал будет представлен в виде текста с изображениями по теме.

Программный продукт должен включать в себя две части:

* обучающую часть, которая должна содержать все необходимые сведения о космосе;
* контролирующую часть, которая должна содержать тест не менее чем из 10 вопросов, на каждый из них должны быть приведены не менее трёх вариантов ответа.

Входными данными в приложении будут вопросы теста и варианты ответов, которые должны храниться в БД, файлы с информацией по данной теме, выбранные варианты ответов.

Выходными данными в приложении будут информация, вопросы и варианты ответов на них, вывод количества правильных ответов и оценка за тест.

Приложение будет состоять из форм, на которых будет располагаться вся необходимая пользователю информация.

Способ решения поставленных задач зависит от выбора средств разработки.

Тестирование программы будет проводится путем выбора пунктов меню и проверки корректности получаемого отклика программы.

Блок тестирования будет проверяться путем запуска теста и эмуляцией различных вариантов пользовательских действий.

## 1.3 Требования к программному обеспечению и техническим средствам

Для корректной работы разрабатываемой программы необходимо, чтобы компьютер удовлетворял следующим минимальным требованиям:

* процессор Pentium J3710;
* объем свободной оперативной памяти не менее 256 Мб;
* видеоадаптер Intel HD Graphics 405;
* 100 Мб свободного места на жестком диске;
* стандартная клавиатура с 101/109 клавишами;
* манипулятор типа «мышь».

Для нормального функционирования системы на компьютере должна быть установлена операционная система Windows 7/8/8.1/10, СУБД Microsoft SQL Server и Windows Media Player.

## 1.4 Проектирование функциональности и структуры программы

Для описания основных функций проектируемой обучающей и контролирующей программы воспользуемся методологией SADT с использованием диаграмм IDEF0.

Сначала строим самую общую, контекстную диаграмму (рисунок 1). На ней описываем процесс обучения и контроля с помощью программного приложения одним функциональным блоком, определяем входные и выходные данные, управляющие воздействия и механизмы.

Входные данные:

* информация по теме (I1);
* тест, обеспечивающий контроль знаний (I2).

Выходные данные:

* отображенная информация (O1);
* количество правильных ответов в тесте (O2);
* оценка за выполненный тест (O3).

Управление:

* дидактические особенности составления материалов для обучения (C1);
* методические рекомендации по составлению теста (C2).

Механизмом будет являться создаваемое приложение (M1).

Субъект: приложение для обучения и контроля знаний.

Точка зрения: процесс обучения и контроля знаний рассматривается с точки зрения корректного визуального отображения текстовой и графической информации, текста вопроса и его вариантов ответа, а также правильного расчета оценки за выполненный тест.

Цель: рассмотрение функций системы, выполнение которых позволит корректно отобразить текстовую и графическую информацию, а также осуществить проверку знаний с помощью теста, с выставлением правильно рассчитанной оценки.

O1

**Обучение и**

**контроль**

С1

С2

I1

I2

M1

**A00**

O2

O3

**Top**

M1 – приложение

I1 – информация по теме

I2 – тест

O1 – отображенная информация

O2 – количество правильных ответов

O3 – оценка за выполненный тест

C1 – дидактические особенности составления материалов для обучения

С2 – методические рекомендации по составлению теста

Рисунок 1 – Контекстная IDEF0-диаграмма

Далее производим декомпозицию функционального блока A00 (рисунок 2), выделяя подфункции «Обучение» – блок A10, и «Контроль» – блок A20.

**A20**

O1

**Обучение**

С1

С2

I1

I2

M1

**A10**

O2

O3

**A00**

**Контроль**

Рисунок 2 – IDEF0-диаграмма основных процессов обучающей и контролирующей программы

На вход функции «Обучение» будет подаваться информация по теме из файлов (I1). Информация будет преобразовываться в соответствии с дидактическими особенностями составления учебных материалов (С1) и выдаваться пользователю.

Для блока «Контроль» входными данными будут материалы теста (I2). Выходными данными будут количество правильных ответов (O2) и оценка за выполненный тест (O3).

Обеспечивать выполнение функций «Обучение» и «Контроль» будет разрабатываемое приложение.

Теперь на отдельных диаграммах произведем декомпозицию функциональных блоков «Обучение» и «Контроль».

На диаграмме A10 (рисунок 3) представлены основные процессы, позволяющие корректно отобразить информацию по выбранной теме.

**Обучение**

**A20**

O1

**Отобразить текстовую информацию**

С1

I1

I2

M1

**A10**

**A10**

**Отобразить графическую информацию**

текст

графика

Рисунок 3 – IDEF0-диаграмма основных процессов обучения

На диаграмме A20 (рисунок 4) представлены основные функции, осуществляемые проектируемым приложением, которые позволят корректно провести тестирование:

* «Отобразить текст вопроса» (A21);
* «Отобразить варианты ответа» (A22);
* «Считать выбранный вариант ответа» (A23);
* «Проверить правильность ответа» (A24);
* «Рассчитать оценку за тест» (A25).

**Отобра-зить текст вопроса**

С2

I2

M1

**A21**

O2

O3

**A20**

**отобразить варианты ответа**

**A22**

**считать выбранный вариант ответа**

**A23**

**проверка правиль-ности ответа**

**A24**

**расчёт оценки за тест**

**A25**

**Обучение**

Рисунок 4 – IDEF0-диаграмма основных процессов контроля

Тест (I2) будет входными данными для первого блока A21. Выбранный вариант ответа будет выходным данным для блока A23 и входным для блока A24. Для блока A25 входными данными будут баллы за ответы.

Выходными потоками для блока A25 будут:

* количество правильных ответов/количество набранных баллов(O2);
* оценка за выполненный тест (O3).

Результатом выполнения блока «Контроль» будут количество баллов за пройденный тест и оценка, которая зависит от количества набранных баллов.

С целью удостоверения предъявляемых к программе требований, приведенных в разделе 1.2, разрабатываемое приложение будет состоять из следующих окон:

* загрузки – будет запускаться перед отображением главного окна;
* главное – будет содержать навигационные кнопки, для перехода на окно тестирования, а также содержать теоретический материал;
* обучения – в нем будет отображаться теоретический материал;
* тестирования – в нем будут отображаться вопросы и варианты ответов к ним;
* вывода результатов – будет служить для вывода результата тестирования;
* справки – будет выводить справочный материал для пользователя.

Переход к блоку тестирования будет происходить при выборе пункта «Тестирование» в главном меню.

Материалы теста будут хранится в БД. Проектируемая структура приложения представлена на рисунке 5.

Окно загрузки

Главное окно

Окно тестирования

Окно справки

Окно результата теста

БД

Рисунок 5 – Схема проектируемой структуры приложения

Тестирование программы планируется проводить путём выбора каждого пункта меню и проверки корректности получаемого отклика программы. Блок тестирования будет проверяться путем запуска тестов и эмуляцией различных вариантов пользовательских действий.

## 1.5 Разработка концептуальной модели базы данных

Для хранения материалов теста будет разработана БД. Она будет состоять из таблицы «Вопросы и ответы», которая включает в себя следующие поля:

* Id – служебное поле для обеспечения уникальности записей;
* Question – содержит текст вопроса;
* Answer1– содержит первый вариант ответа на вопрос в текстовом виде;
* Answer2 – содержит второй вариант ответа на вопрос в текстовом виде;
* Answer3 – содержит третий вариант ответа на вопрос в текстовом виде;
* Answer4 – содержит четвертый вариант ответа на вопрос в текстовом виде;
* Right – номер правильного ответа в списке вариантов ответов.

Концептуальная модель БД представлена на рисунке 6.

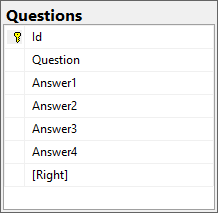


Рисунок 6 – Концептуальная модель БД

## 1.6 Выбор средств разработки

Из множества различных языков программирования (C++, С#, Java, Delphi,) для реализации курсового проекта был выбран C# и интегрированная среда разработки Microsoft Visual Studio 2019.

C# является объектно-ориентированным и в этом плане много перенял у Java и С++. Он поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. C# продолжает активно развиваться, и с каждой новой версией появляется все больше интересной функциональности, как, например, лямбда-выражения, динамическое связывание, асинхронные методы и так далее.

Преимущества C# по сравнению с аналогичными программными продуктами:

* поддержка подавляющего большинства продуктов Microsoft;
* типы данных имеют фиксированный размер что повышает мобильность языка и упрощает программирование;
* автоматическая очистка оперативной память;
* большое количество специальных конструкций, разработанных для понимания и написания кода [7].

Microsoft Visual Studio – полнофункциональная интегрированная среда разработки (IDE) с поддержкой популярных языков программирования, среди которых С, C++, VB.NET, C#, F#, JavaScript, Python. Функциональность Visual Studio охватывает все этапы разработки программного обеспечения, предоставляя современные инструменты для написания кода, проектирования графических интерфейсов, сборки, отладки и тестирования приложений. Возможности Visual Studio могут быть дополнены путем подключения необходимых расширений. Visual Studio поддерживает подсветку синтаксиса, вставку фрагментов кода, отображение структуры и связанных функций. Существенно ускорить работу помогает технология IntelliSense – автозавершение кода по мере ввода.

Встроенный отладчик Visual Studio используется для поиска и исправления ошибок в исходном коде, в том числе на низком аппаратном уровне. Инструменты диагностики позволяют оценить качество кода с точки зрения производительности и использования памяти [8].

Для реализации справочной системы был выбран HTML HelpWorkShop. Этот инструмент имеет легкий для использования и интуитивно понятный графический интерфейс. С помощью него можно создавать проекты, гипертекстовые разделы, файлы содержаний, заголовочные файлы, а также прочие элементы, которые могут понадобиться для последующего использования в справочной системе [10].

Для создания инсталляции проекты был выбран Smart InstallMaker. Это условно-бесплатная утилита с закрытым исходным кодом, которая представляет пользователям мощный и простой в использовании инструмент для создания высокоэффективных инсталляторов в 32-битных операционных системах Microsoft Windows.

Smart InstallMaker обладает богатыми функциональными возможностями для создания надёжных и красивых инсталляционных пакетов. Утилита предоставляет мощный алгоритм Cabinet для сжатия и создания компактных инсталляторов с маленьким размером, поддерживает создание многоязычного инсталлятора (до 25 языков), а также предоставляет средства для управления внешним видом программы-установки. Среди прочих возможностей Smart InstallMaker, можно выделить создание деинсталлятора, ярлыков в указанных директориях системы, регистрацию новых шрифтов или расширений в операционной системе, поддержку ActiveX и переменных, проверку установленного .NET Framework, разделение инсталлятора на установочные диски, детальную настройку информации об установочном файле, системных требований (например, запретить установку программы в Windows 95), а также его внешнего вида.

Недостатки:

* отсутствие кроссплатформенности;
* закрытый исходный код;
* отсутствует поддержка 64-разрядных операционных систем [11].

Microsoft SQL Server – система управления реляционными БД, разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с небольшими и средними по размеру БД, а также крупными базами данных масштаба предприятия. SQL является общепринятым интерфейсом к БД [8].

# 2 РАБОЧИЙ ПРОЕКТ

## 2.1 Проектирование физической модели базы данных

Физическая модель данных описывает то, как данные хранятся в компьютере, представляя информацию о структуре записей, их упорядоченности и существующих путях доступа. Также описываются типы, идентификаторы и разрядность полей.

На основе логической модели были создана физическая модель БД, которая хранится в компьютере в файле SpaceTest.mdf (таблица 1).

Таблица 1 – Физическая модель таблицы «Вопросы и ответы»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Подпись поля | Тип данных | Длина | Ключ |
| Id | Номер вопроса | int | - | Первичный |
| Question | Вопрос | nvarchar | 255 | - |
| Answer1 | Первый ответ | nvarchar | 255 | - |
| Answer2 | Второй ответ | nvarchar | 255 | - |
| Answer3 | Третий ответ | nvarchar | 255 | - |
| Answer4 | Четвертый ответ | nvarchar | 255 | - |
| Right | Верный ответ | int | - | - |

## 2.2 Функциональное взаимодействие модулей программы

На основе разработанного технического проекта было написано приложение по теме «Космос». Для создания программы был использован язык программирования C# и ИСР Visual studio 2019. В программе используются следующие языковые модули:

* System – содержит фундаментальные и базовые классы;
* System.Windows.Forms – содержит классы для создания работы с формами;
* System.Media – содержит классы для воспроизведения звуковых файлов и работы с системными звуками;
* System.Drawing – предоставляет доступ к основным графическим функциям GDI+ (Graphics Device Interface);
* System.Data.SqlClient – содержит классы для работы с SQL Server [1].

Также были разработаны следующие пользовательские модули, взаимодействие которых представлено на рисунке 7:

* MainForm.cs (главное окно программы) – осуществляет переход между режимами обучения и тестирования, а также обеспечивает вывод руководства по использованию программы;
* TestForm.cs (окно тестирования) – обеспечивает интерфейс для тестирования: отображения текста вопроса и вариантов ответа, переход к следующему вопросу, расчет оценки;
* RatingForm.cs (окно результатов тестирования) – обеспечивает интерфейс для вывода результатов тестирования;
* Zastavka.cs (окно загрузки) обеспечивает отображение картинки приветствия;
* Help.chm (справка) – обеспечивает отображения справочной информации о программе и работе с ней.
* Question.cs (класс вопросов) – обеспечивает создания объектов вопроса и ответов для теста, а также содержит статистическую переменную для хранения ссылки на подключение к БД (приложение А).

Окно загрузки (Zastavka)

Главное окно

(MainForm)

Окно тестирования

(TestForm)

Окно справки (Help)

Окно результата теста

(RatingForm)

Класс вопросов

(Question.cs)

Рисунок 7 – Схема взаимодействия основных модулей программы

В модуле MainForm.cs описаны и используются следующие процедуры:

* private void солнечнаяСистемаToolStripMenuItem\_Click(object sender, EventArgs e) – открывает текстовую область, с текстом «Солнечная система»;
* private void button1\_Click(object sender, EventArgs e) – воспроизводит звук Солнца;
* private void testMenu\_Click(object sender, EventArgs e) – открывает окно тестирования;
* private void справкаToolStripMenuItem\_Click(object sender, EventArgs e) – открывает окно со справкой;
* private void картыСолнечнойСистемыToolStripMenuItem\_Click(object sender, EventArgs e) – открывает карту Солнечной системы с планетами;
* public void PlanetClick(string link) – открывает текстовую область, с текстом о планете, на которую нажал пользователь.

В модуле TestForm.cs описаны и используются следующие процедуры:

* private void TestForm\_Load(object sender, EventArgs e) – загружает данные для теста при загрузке формы;
* private void Zapolnenie(int index) – отображает вопрос и варианты ответа на экране;
* private void applyAnswer\_Click(object sender, EventArgs e) – принимает ответ пользователя.

В модуле RatingForm.cs описана и используется процедура private void OkButton\_Click(object sender, EventArgs e), позволяющая закрыть окно результатов.

## 2.3 Входные и выходные данные

Входными данными для модуля MainForm.cs являются данные из текстовых документов (.rtf), которые загружаются в mainRichTextBox и картинки (.jpg/.jpeg/.gif), которые загружаются в SunPictureBox, MercuryPictureBox, VenusPictureBox, EarthPictureBox, MarsPictureBox, JupiterPictureBox, SaturnPictureBox, UranusPictureBox, и NeptunepictureBox.

Входными данными для модуля TestForm.cs являются данные из таблицы «Questions» БД SpaceTest.mdf.

Выходными данными для модуля TestForm.cs является переменная int rightAnswers хранящая в себе количество набранных баллов.

## 2.4 Установка и настройка программы

Для установки программы необходимо запустить файл инсталляции Space Travel Install.exe. В результате в выбранном пользователем каталоге будет создана папка «Space Travel», в которой будут размещены исполняемых файл программы (SpaceTravel.exe), файл системы помощи (help.chm), и файлы с материалами (.rtf-файлы, БД, и файлы с изображениями). Также будет создан ярлык для запуска приложения на рабочем столе. Ярлык для деинсталляции программы будет размещен в папке программы.

Для создания инсталляции была использована программа Smart Install Maker. В ходе инсталляции пользователю показываются следующие окна:

* выбор папки установки;
* создание дополнительных ярлыков – позволяет создать ярлык на рабочем столе или отказаться от создания ярлыка;
* ход выполнения процесса установки;
* завершение установки – позволяет выбрать файлы, которые нужно запустить после установки.

После запуска программы инсталляции нужно следовать указаниям в появляющихся окнах.

## 2.5 Работа с программой

При запуске программы отображается окно загрузки (рисунок 8). Через две секунды загружается главное окно (рисунок 9).



Рисунок 8 – Окно загрузки

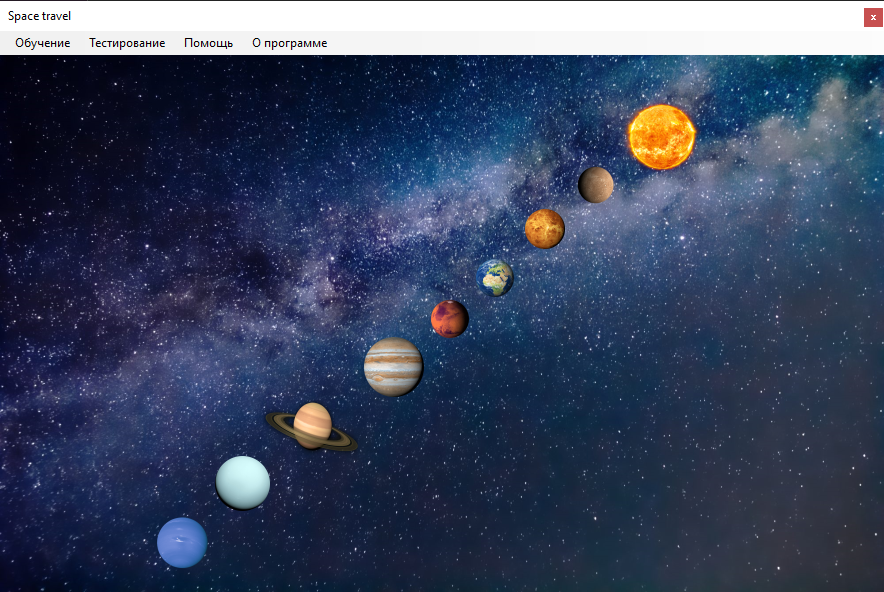


Рисунок 9 – Главное окно

Чтобы прочитать теоретический материал, необходимо навести на пункт меню «Обучение», после чего отобразится список теоретического материала (рисунок 10).

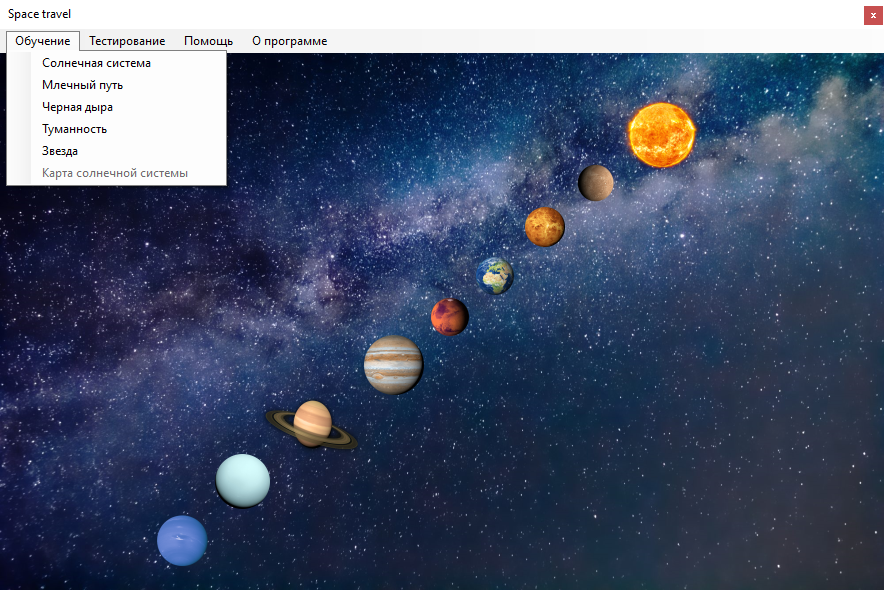


Рисунок 10 – Список теоретического материала

Чтобы пройти тестирование, следует нажать на пункт меню «Тестирование», после чего появится форма, содержащая в себе первый вопрос и четыре варианта ответа (рисунок 11).

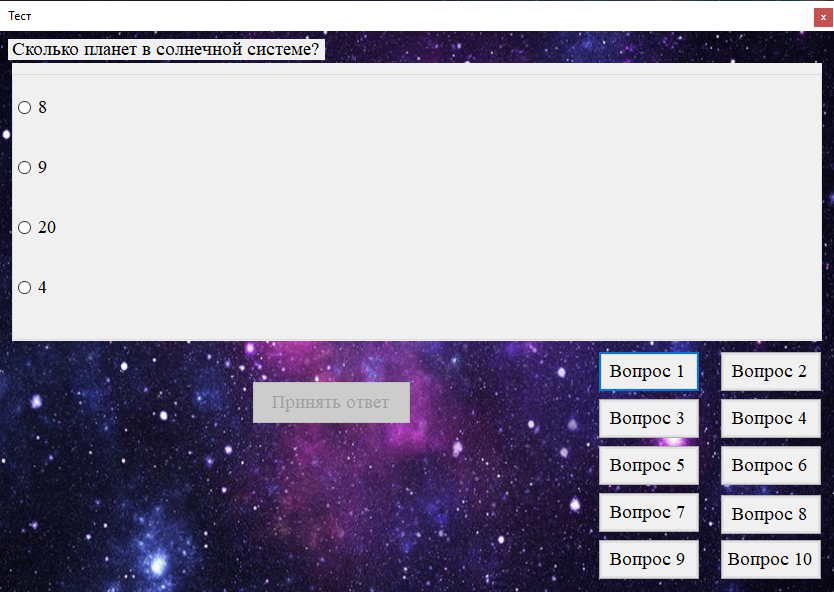


Рисунок 11 – Окно теста

Для перехода к следующему вопросу, можно выбрать ответ на текущий вопрос и подтвердить ответ, или использовать кнопки вопросов. После того как будут получены ответы на все вопросы, будет доступна кнопка «Завершить тест», которая вызывает функцию подсчёта ответов, и выдаёт результат в окне с результатами тестирования (рисунок 12).

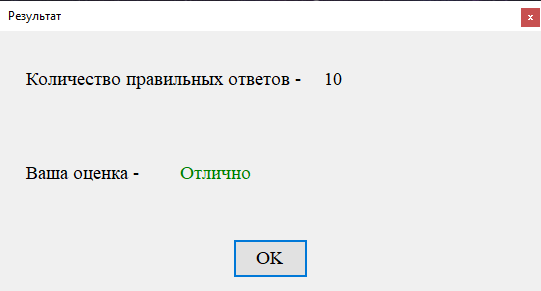


Рисунок 12 – Окно теста

Для получения справочной информации требуется нажатие на кнопку «Помощь» (рисунок 13).

Для создания справочной системы использовалась программа HTML HelpWorkshop. Система помощи включает в себя следующие страницы:

* Главная форма – содержит основные сведения о главной форме;
* Меню – содержит сведения о меню в главной форме;
* Форма тестирование – здесь можно прочитать о том, как нужно проходить тест, как отвечать на вопросы;
* Элементы управления – содержит описания кнопок управлением теста.

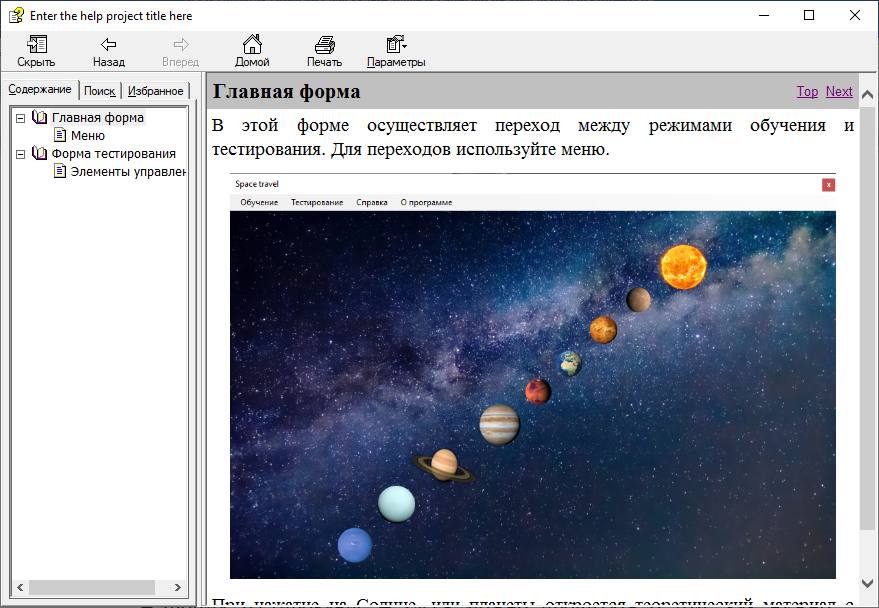


Рисунок 13 – Окно теста

# ЗАКЛЮЧЕНИЕ

В результате выполнения данного курсового проекта было написано приложение по теме «Космос», с помощью которой можно легко и быстро освоить достаточно обширный материал.

Основными достоинствами программы являются:

* программа имеет простую установку, настройку и эксплуатацию;
* наличие удобного графического интерфейса;
* выставление оценки с указания количества правильных ответов;
* данный программный продукт будет предназначен для широкого круга пользователей;
* наличие справочной системы;
* развитие памяти.

Использование данной программы избавит от необходимости:

* составлять тест вручную;
* подсчитывать результаты теста;
* определять оценку.

В ходе выполнения курсового проекта были достигнуты поставленные цели и решены следующие задачи:

* исследованы основные объекты космического пространства;
* спроектированы структура приложения и БД;
* разработаны справочные и тестовые материалы;
* определены минимальные требования к техническим и программным средствам;
* описаны входные и выходные данные;
* выбраны средства реализации приложения;
* разработана БД;
* разработан программный код приложения;
* создано руководство пользователя;
* создана инсталляция.

К недостаткам данной программы можно отнести то, что отсутствует возможность сохранения результатов тестирования.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Албахари Б., Албахари Дж. С# 7.0. Справочник. Полное описание языка. М.: Вильямс, 2018. 1024 с.
2. Виссер Д. Разработка обслуживаемых программ на языке С#. М.: ДМК Пресс, 2017. 194с.
3. Грофф Дж., Вайнберг П., Оппель Э. SQL Полное руководство. 3-е изд. СПб.: Диалектика, 2019. 962 с.
4. Джепикс Ф., Троелсен Э. Язык программирования C# 7 и платформы .NET и .NET Core. М.: Вильямс, 2018. 1328 с.
5. Кугаевских А. Проектирование информационных систем. Системная и бизнес-аналитика. Новосибирск: НГТУ, 2018. 256 с.
6. Мюллер Д. П., Семпф Б., Сфер Ч. С# Для чайников. М.: Диалектика, 2016. 608 с.
7. Скит Д. С# Для профессионалов. Тонкости программирования. М.: Вильямс, 2019. 608 с.
8. Документация Microsoft. URL: https://docs.microsoft.com/ru-ru/documentation (дата обращения: 11.11.2020).
9. Космос Планет. Сайт всё о космосе. URL: https://cosmosplanet.ru/ solnechnayasistema/stroenie-solnechnoj-sistemy.html (дата обращения: 12.11.2020).
10. Официальный сайт HTML HelpWorkShop. URL: https:// www.microsoft.com (дата обращения: 10.12.2020).
11. Официальный сайт Smart InstallMaker. URL: <http://ru.sminstall.com> (дата обращения: 10.12.2020).

ПРИЛОЖЕНИЕ А

Листинг модуля Question.cs

class Question.cs

{

static public string ConnectionString

{

get

{return "Data Source = DESKTOP-OI45NF6; Initial Catalog= SpaceTest; Integrated Security=True"; }

}

public string Questio { get; set; }

public string Option1 { get; set; }

public string Option2 { get; set; }

public string Option3 { get; set; }

public string Option4 { get; set; }

public int rightOption { get; set; }

public Question()

{

this.Questio = "";

this.Option1 = "";

this.Option2 = "";

this.Option3 = "";

this.Option4 = "";

this.rightOption = 0;

}

}

public void SizeUp(object sender, EventArgs e)

{

PictureBox pic = sender as PictureBox;

pic.Width += 15;

pic.Height += 15;

pic.Left -= 7;

pic.Top -= 7;

}

public void SizeDown(object sender, EventArgs e)

{

PictureBox pic = sender as PictureBox;

pic.Width -= 15;

pic.Height -= 15;

pic.Left += 7;

pic.Top += 7;

}

public void PlanetClick(string link)

{

mainPanel.Visible = true;

SolarSystemPanel.Visible = false;

mainRichTextBox.LoadFile(link);

картыСолнечнойСистемыToolStripMenuItem.Enabled = true;

}

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

{

SqlConnection sqlConnection = new SqlConnection();

sqlConnection.ConnectionString = Question.ConnectionString;

SqlCommand sqlCommand = new SqlCommand("exec TakeInfo", sqlConnection);

sqlConnection.Open();

SqlDataReader sqlDataReader = sqlCommand.ExecuteReader();

int i = 0;

while (sqlDataReader.Read())

{

answer[i] = 0;

questions[i] = new Question();

questions[i].Questio = sqlDataReader[1].ToString();

questions[i].Option1 = sqlDataReader[2].ToString();

questions[i].Option2 = sqlDataReader[3].ToString();

questions[i].Option3 = sqlDataReader[4].ToString();

questions[i].Option4 = sqlDataReader[5].ToString();

questions[i].rightOption = Convert.ToInt32(sqlDataReader[6].ToString());

i++;

}

sqlConnection.Close();

option1.Checked = false;

}

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

{

rightAnswers = 0;

for (int i = 0; i < 10; i++) if (answer[i] == questions[i].rightOption)

rightAnswers++;

RatingForm ratingForm = new RatingForm();

ratingForm.rightAnswers = rightAnswers;

ratingForm.ShowDialog();

closeTestButton.Visible = true;

}

void OpenFinishButton()

{

for (int i = 0; i < 10; i++) if (answer[i] == 0) return;

finishTest.Visible = true;

}