МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления



**ОТЧЁТ**

**лабораторной работе №3**

**по теме:** Разработка приложения (Текстовый квест)

**по дисциплине:** Программирование.

Вариант **№5**

Выполнили работу:

Студенты гр. АП-227, АВТФ

Камаев М.С.

Патотов Н.М.

Лузянин З.С.

Проверил работу:

Старший преподаватель кафедры АСУ Эстрайх И.В.

«\_\_\_» \_\_\_\_\_\_ 20\_\_г

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Новосибирск 2023

**Оглавление**

**Постановка задачи**………..………………………………………………………………………………………………………………………3

**Тестовые данные**……………………………………..……………………………………………………………………………………………4

**Описание разработанных классов**……………………………………………………………………………………………………….6

**Тестирование программы**……………………………………………………………………………………………………………………7

**Оценка соответствия полученных результатов и теоретических представлений**…………………………8

**Выводы**……………………………………………………………………………………………………………………………………………..…9

**Исходный код программы**…………………………………………………………………………………………………………………10

**Постановка задачи**

Вы должны работать в команде. Для этого начните с проектирования будущего приложения, разделите классы между членами бригады, договоритесь об интерфейсах, через которые ваши классы будут взаимодействовать. Исходный код должен находиться в репозитории Git, быть покрыт модульными тестами и документирован. Объекты должны поддерживать сериализацию и десериализацию. Должна быть возможность закрыть программу с сохранением состояния и затем запустить её из сохранённого состояния. ГПИ в этой ЛР не нужен.

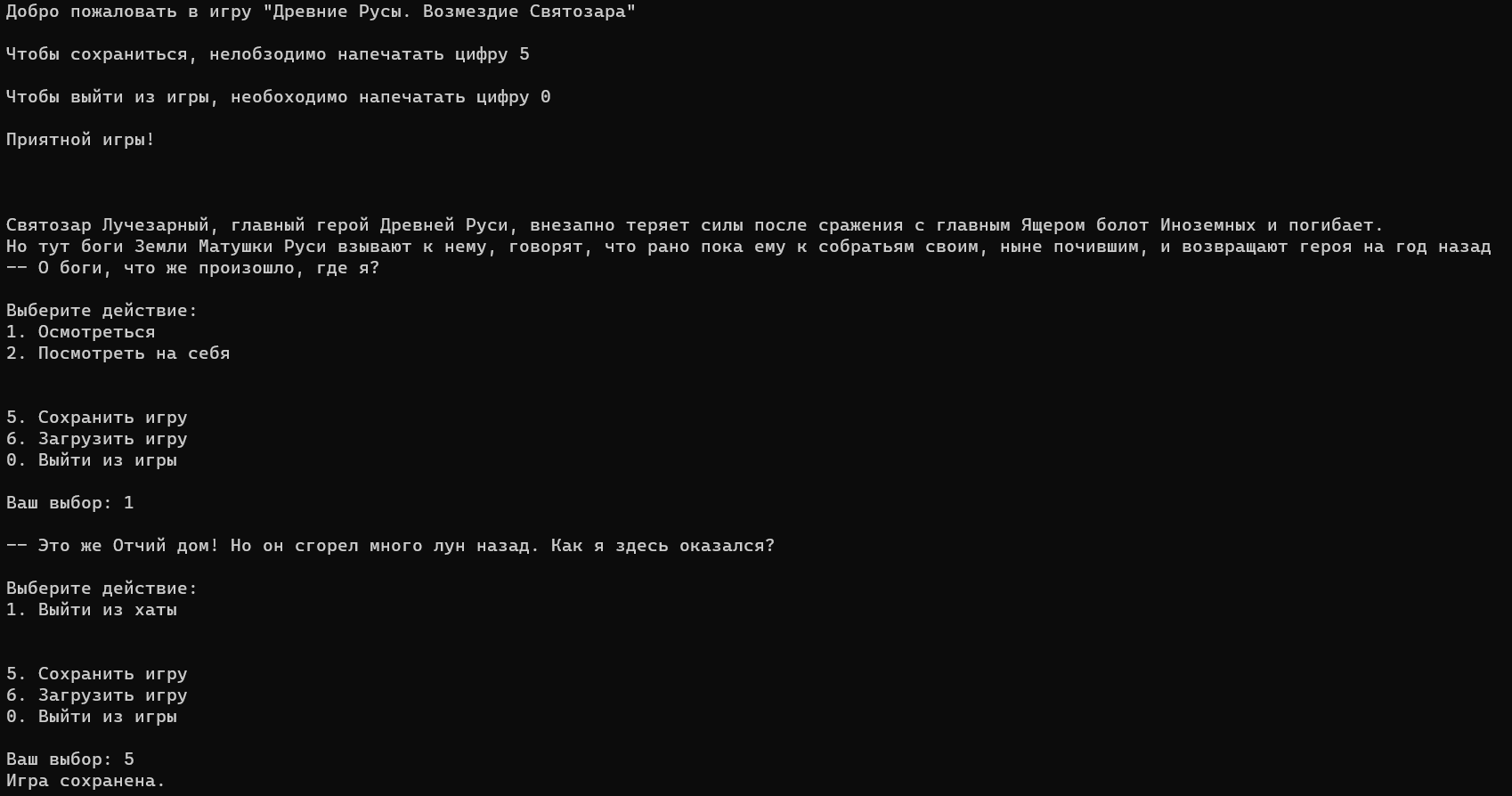
**Описание разработанных классов**

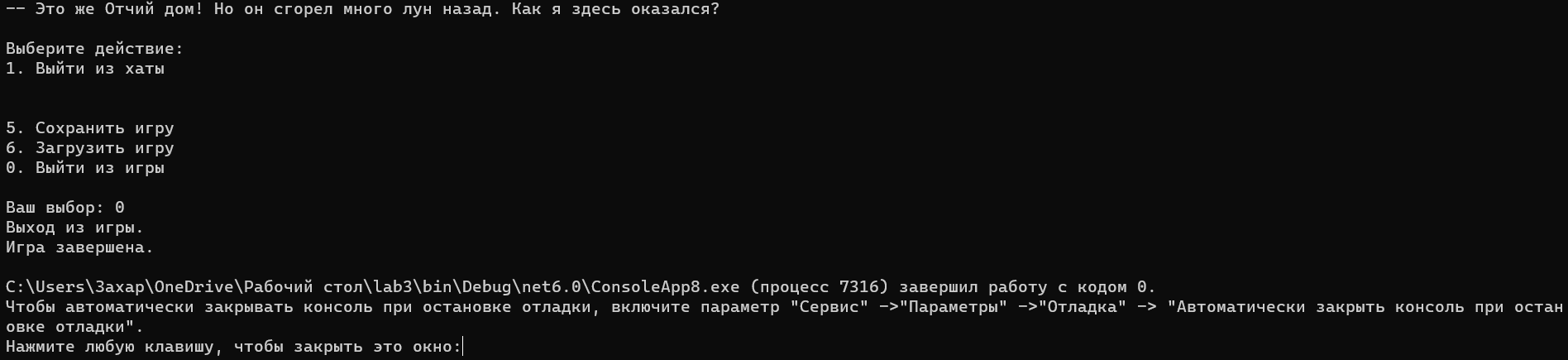
|  |  |  |
| --- | --- | --- |
| Классы | Поля | Методы |
| Class Choice | public string Description  public int NextRoom | public Choice(string description, int nextRoom) |
| class Game | public Player Player  public Room[] Rooms | public Game(Player player, Room[] rooms) |
| Class Player | public string Name  public int CurrentRoom | public Player(string name, int currentRoom) |
| Class Program | Player player  Room[] rooms | static void PlayGame(Game game)  static void SaveGame(Game game)  static void LoadGame(out Game game) |
| Class Room | public string Description  public List<Choice> Choices | public Room(string description)  public void AddChoice(string description, int nextRoom) |

# 

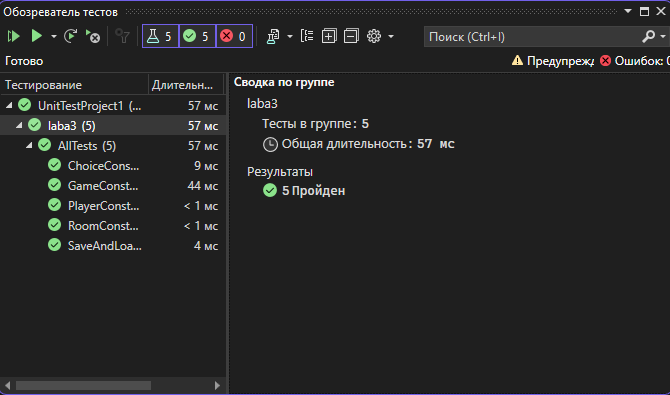
# **Тестирование программы**

**Часть 1**

****

****

**Модульный тест**

****

**Оценка соответствия полученных результатов и теоретических представлений:** Полученные результаты точно соответствуют теоретическим представлениям.

# **Выводы**

В ходе лабораторной работы мы написали код для Текстового квеста, который поддерживает сериализацию в файл, формата .dat .Однако, в ходе лабораторной работы модульные тесты выполнены не были из-за непредвиденной ошибки компилятора.

# 

# **Исходный код программы**

**Choice.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp8

{

[Serializable]

class Choice

{

public string Description { get; set; }

public int NextRoom { get; set; }

public Choice(string description, int nextRoom)

{

Description = description;

NextRoom = nextRoom;

}

}

}

**Game.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp8

{

[Serializable]

class Game

{

public Player Player { get; set; }

public Room[] Rooms { get; set; }

public Game(Player player, Room[] rooms)

{

Player = player;

Rooms = rooms;

}

}

}

**Player.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp8

{

[Serializable]

class Player

{

public string Name { get; set; }

public int CurrentRoom { get; set; }

public Player(string name, int currentRoom)

{

Name = name;

CurrentRoom = currentRoom;

}

}

}

**Room.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp8

{

[Serializable]

class Room

{

public string Description { get; set; }

public List<Choice> Choices { get; set; }

public bool IsSolved { get; set; }

public Room(string description)

{

Description = description;

Choices = new List<Choice>();

IsSolved = false;

}

public void AddChoice(string description, int nextRoom)

{

Choices.Add(new Choice(description, nextRoom));

}

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Runtime.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

namespace ConsoleApp8

{

class Program

{

static void Main()

{

Console.WriteLine("Добро пожаловать в игру \"Древние Русы. Возмездие Святозара\"\n\nЧтобы сохраниться, нелобзодимо напечатать цифру 5\n\nЧтобы выйти из игры, необоходимо напечатать цифру 0\n\nПриятной игры!\n\n");

// Инициализация игры

Player player = new Player("Игрок", 1);

Room[] rooms = {

//0

new Room("\n\n\n\nВы Умерли...\n\n\n\n"),

//1

new Room("\nСвятозар Лучезарный, главный герой Древней Руси, внезапно теряет силы после сражения с главным Ящером болот Иноземных и погибает.\nНо тут боги Земли Матушки Руси взывают к нему, говорят, что рано пока ему к собратьям своим, ныне почившим, и возвращают героя на год назад\n-- О боги, что же произошло, где я?"),

//2

new Room("\n-- Это же Отчий дом! Но он сгорел много лун назад. Как я здесь оказался?"),

//3

new Room("\n-- Да и шрамов боевых меньше стало... Как же так? Что за проделки черта?"),

//4

new Room("\n-- Солнце будто ярче светит, хотя может мне кажется... \n-- Свяяяяят! Иди сюда! Ну же, скорее!\nСвятозар слышит голос, до боли знакомый. Брат... Родной... Но его же убил при мне Главный Ящер...\n Святозар, еле сдерживая слезы, бежит к брату, обнимает его.\nОт такого внезапного действия Всеволод пошатнулся, но удержался на ногах. Приняв объятия, Всеволод спрашивает:\n-- Братец, родной, что же ты плачешь? Нежели обидел тебя кто? \n-- Нет, брат, удивился я сильно, увидев тебя! Ты же умер от него..."),

//5

new Room("\n-- Успокойся же! Пойдем лучше в корчму меду выпьем! \n Но как же ящеры? Нам нужно срочно их изнечтожить, пока они нас всех не погубили!\nКакие еще ящеры, братец? Неужто приснился сон тебе кошмарный? Сходика лучше ты к Марье, она поможет тебе! \n "),

//6

new Room("\n-- Марья, подруга сердечная Я К те...\n -- Знаю, Святозарушка, зачем ко мне ты пришел... Ворожба сказала мне, что придет муж Руси Матушки ко мне, помочь ему должна я...\nИ знаю я как. Сходи, Святозар, за Тьмутаракань, в горы высокие. Там, глубоко в горах, прячутся твари иноземные, истреби их.\nНе забудь дружину собрать, иначе костей не соберешь! \n-- Спасибо за указ, бабушка, последую ему я.\n-- Удачи тебе, пусть Перун благоволит тебе!"),

//7

new Room("\nСвятозар заходит в корчму, видит, как десяток богатырей смотрят за битвой на кулаках между братом Всеволодом и каким-то знакомым витязем.\nСвятозар присоединяется к просмотру.\n --ВСЕ-ВО-ЛОД!\n--ВСЕ-ВО-ЛОД!\nВсеволод с размаху ударяет прямо в челюсть противнику, бой заканчивается...\n"),

//8

new Room("\n-- Ха-ха-ха-ха! Всем мёду от меня!\n--Неужто ты, Всеволод, молодняку решил класс показать?\n--Конечно! Ну а как их еще научить уму разуму, брат?\n--И то верно..."),

//9

new Room("\nСвятозар с Всеволодом выпивают глоток за глотком мед, будто бы соревнуясь. Кружка уже пуста."),

//10

new Room("\n-- Всеволод, у меня срочно дело есть. За Тьмутараканью, в горах высоких, живет нечисть иноземная, ящерами зовутся.\nСам Перун велел нам их погубить, Ворожея нас уже благословила. Осталось только дружину собрать.\n-- Это, здорово, хоть какое-то веселье! Есть у меня ребята здравые, человек 10 соберем, но нужно все же подумать, кого из них взять в силу главную...\n-- Вот гляди - есть тот парень, что со мною бился. Да, молодой он, сильный, храбрый, но неопытный. Да и в настоящем бою научится как биться нужно.\nСпроси его, но помни, слаб он пока для такого рода дел. \n-- Есть еще дядька наш Василевс. Хоть и стар он уже, но вояка бывалый. Опыт богатый, мудрый он, но все же слаб, да и слушаться нас не будет, все ему по своему сделать надо.\n-- Спасибо брат, подумаю я еще тогда..."),

//11

new Room("\nСвятозар выпивает еще кружку меду, и думает, кого бы взять\n-- Малец, хоть и не опытный, но послушный будет, силы ему точно не занимать, даже Всеволода чуть не победил, но, боюсь, Ящеры мудрее будут, да обманут его, а он и не заметит...\r\n\tСтарший же, скорее всего, помогать нам в бою не будет, но может помочь тактикой. Сломим ящеров, а они и не поймут!\n"),

//12

new Room("\n-- Друг мой! Призываю тебя я в бой с нечистью инородной! В горы, за Тьмутаракань пойдем! Пойдешь ли ты с дружиною нашей? Опыту наберешься, да и силу свою испробуешь.\nНо смотри, опасно это будет. Ящеры сила злобная, неизвестная, так и убить могут тебя в битве!\n-- Конечно пойду, старший! Как раз кулаки чесались!\n-- Тогда собирайся, выдвигаемся на рссвете, идти долго будем..."),

//13

new Room("\nВы пришли в Тьмутаракань...\n\nГород полностью разрушен, на улице нет признака жизни...\n\n-- Вот она какая, Тьмутаракань... Точнее была...\nВы прислушиваетесь, что-то бежит от вас за домом слева... Ящер!\n-- Братья! Изловить шпиона вражеского!\nВы и Ваша дружина бежите за мелким ящером, но он вбегает в закоулок, где вас ждут 3 средних Ящера.\n-- К бою, братцы!"),

//бой

//14

new Room("\nВы поймали ящера разведчика..."),

//15

new Room("\n-- Колись, супостат! Где же град ваш? А ежели не скажешь, то замучаем тебя, но убивать не будем, заставим жить, терзая...\n--Смилуйся, богатырь... проведу я вас в град наш. Но послушай же, трудно нам приходится. Земля наша иссохла, урожая нет уже годами, народ наш мучается.\r\n\tЗемли для отроков наших больше нет. Только, захватывая, можем мы прожить чуть дольше.\n-- то ты мне тут рассказываешь, мелочь, лучше расскажи да покажи куда идти к главному вашему, с ним разговаривать буду!\n-- Тогда идем, витязь!"),

//16

new Room("\nВы пришли в поселение Ящеров..."),

//17

new Room("\nВокруг сидят семьи ящеров вокруг костров, разруха.\nВас отводят к главному ящеру, который сидел в крупном шатре, в окружении пяти ящеров-генералов"),

//18

new Room("\n-- Дружинники! Рад приветствовать у себя! Зачем же вы проделали такой путь, храбрые воины? Убить ли меня вы хотите или беседу вести будете?\n-- Это будет зависить от того, что ты расскажешь нам, ежели опасность почуем - мигом меч у горла поставлю!\n-- Полно вам, ребята! Горе у нашего народа случилось, Топи наши родимые иссохли совсем, погибли товарищи наши, от голода. \nОт того и пришлось нам прийти в ваши земли. Простите нас и наши деяния!\n-- Старший! Не надо убивать их! Давайте лучше прогоним их, пусть земли врагов наших забирают, наши же останутся нам!\n-- Как же так, младший! Нам же потом аукнется это! Но может мы и сможем пользу извлечь из этого..."),

//19

new Room("\n-- Не могу я позволить грабить наши земли, ни сейчас, ни потом! Убить их всех!"),

//20

new Room("\n-- Слава Перуну! Победа Наша!"),

//21

new Room("\nПосле победы над Ящерами храбрая дружина вернулась домой и устроила пир на всю Русь Матушку. Выпивая литрами мёд Святозар вдруг задумался, что было бы, оставив он Ящеров в живых...\nМожет, все же не стоило устраивать резню...\nНо теперь это уже не узнать...\n\n\nКонец?\n\n\n"),

//22

new Room("\n-- Пусть это и непростой выбор, но душа моя болит! Живите, иноземцы, но с нашей земли уйдите! Вон, в земли половцев идите! Там земли вам хватит, а на наши не возвращайтесь!\n-- Благодарим тебя, Святозар! От всего племени!\n-- Идите же!"),

//23

new Room("\nПосле того, как вы решили пощадить Ящеров, вы вернулись домой, но чувствовали, что то решение было неправильным...\nВдруг они правда вернутся?\nВдруг они объединятся с половцами?\nНадо готовиться к худшему, но пока нужно дать душе отдохнуть...\n\n\nКонец?\n\n\n"),

//24

new Room("\n-- Не могу я младших в такую битву брать, не дорос он еще...\n\nВы подошли к Василевсу.\n\n-- Брат! Айда в горы, за Тьмутаракань! Земли наши надо защитить от иродов поганых!\n-- Вот еще! Да чтобы младшие мной коммандовали?! Да ни в жизнь!\n-- Друг мой, одумайся! Жизнь всей Руси на волоске!\n-- Пусть так, но не верю вам я, мелочь! Не пойду я с вами!\n-- Что же, придется нам идти в меньшенстве..."),

};

rooms[0].AddChoice("Начать заново", 1);

rooms[1].AddChoice("Осмотреться", 2);

rooms[1].AddChoice("Посмотреть на себя", 3);

rooms[2].AddChoice("Выйти из хаты", 4);

rooms[3].AddChoice("Выйти из хаты", 4);

rooms[4].AddChoice("Пойти к Ворожеи", 5);

rooms[5].AddChoice("Пойти в корчму", 6);

rooms[6].AddChoice("Далее...", 7);

rooms[7].AddChoice("Далее...", 8);

rooms[8].AddChoice("Далее...", 9);

rooms[9].AddChoice("Далее...", 10);

rooms[10].AddChoice("Далее...", 11);

rooms[11].AddChoice("Выбрать Малого", 12);

rooms[11].AddChoice("Выбрать Старшего", 24);

rooms[12].AddChoice("Выйти в поход...", 13);

rooms[13].AddChoice("Хук в лицо", 0);

rooms[13].AddChoice("Призыв молний Перуна", 0);

rooms[13].AddChoice("Славянский зажим в тиски", 14);

rooms[14].AddChoice("Далее...", 15);

rooms[15].AddChoice("Пойти с ящером-разведчиком", 16);

rooms[16].AddChoice("Далее...", 17);

rooms[17].AddChoice("Далее...", 18);

rooms[18].AddChoice("Пощадить", 22);

rooms[18].AddChoice("Убить их!", 19);

rooms[19].AddChoice("Окропить водой Байкальской!", 0);

rooms[19].AddChoice("Славянский зажим в тиски", 0);

rooms[19].AddChoice("Призыв молнии Перуна", 0);

rooms[19].AddChoice("Глубокое проникновение копья", 20);

rooms[20].AddChoice("Далее...", 21);

rooms[22].AddChoice("Далее...", 23);

rooms[24].AddChoice("Выйти в поход", 13)

;

Game game = new Game(player, rooms);

// Запуск игры

PlayGame(game);

Console.WriteLine("Игра завершена.");

}

static void PlayGame(Game game)

{

while (game.Player.CurrentRoom < game.Rooms.Length)

{

Room currentRoom = game.Rooms[game.Player.CurrentRoom];

Console.WriteLine(currentRoom.Description);

if (!currentRoom.IsSolved)

{

Console.WriteLine("\nВыберите действие:");

for (int i = 0; i < currentRoom.Choices.Count; i++)

{

Console.WriteLine($"{i + 1}. {currentRoom.Choices[i].Description}");

}

Console.WriteLine("\n");

Console.WriteLine("5. Сохранить игру");

Console.WriteLine("6. Загрузить игру");

Console.WriteLine("0. Выйти из игры");

Console.Write("\nВаш выбор: ");

string input = Console.ReadLine();

if (input == "0")

{

// Выход из игры

Console.WriteLine("Выход из игры.");

return;

}

else if (input == "5")

{

// Сохранение игры

SaveGame(game);

}

else if (input == "6")

{

// Загрузка игры

Game loadedGame;

LoadGame(out loadedGame);

if (loadedGame != null)

{

// Продолжаем игру с загруженным состоянием

PlayGame(loadedGame);

}

}

else if (int.TryParse(input, out int choiceIndex) && choiceIndex >= 1 && choiceIndex <= currentRoom.Choices.Count)

{

game.Player.CurrentRoom = currentRoom.Choices[choiceIndex - 1].NextRoom;

}

else

{

Console.WriteLine("Некорректный ввод. Попробуйте еще раз.");

}

}

else

{

Console.WriteLine("Вы уже решили эту комнату. Переходите к следующей.");

game.Player.CurrentRoom++;

}

}

}

//функция сохранения игры

static void SaveGame(Game game)

{

try

{

using (FileStream fs = new FileStream("savegame.dat", FileMode.Create))

{

IFormatter formatter = new BinaryFormatter();

formatter.Serialize(fs, game);

}

Console.WriteLine("Игра сохранена.");

}

catch (Exception ex)

{

Console.WriteLine("Ошибка при сохранении игры: " + ex.Message);

}

}

//функция загрузки игры

static void LoadGame(out Game game)

{

game = null; // Инициализация, чтобы избежать ошибки компилятора

try

{

using (FileStream fs = new FileStream("savegame.dat", FileMode.Open))

{

IFormatter formatter = new BinaryFormatter();

game = (Game)formatter.Deserialize(fs);

}

Console.WriteLine("Игра успешно загружена.");

}

catch (Exception ex)

{

Console.WriteLine("Ошибка при загрузке игры: " + ex.Message);

}

}

}

}

**Модульные тесты**

**UnitTest1.cs**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.IO;

using System.Runtime.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

namespace laba3

{

[TestClass]

public class AllTests

{

[Serializable]

public class Choice

{

public string Description { get; set; }

public int NextRoom { get; set; }

public Choice(string description, int nextRoom)

{

Description = description;

NextRoom = nextRoom;

}

}

[Serializable]

public class Player

{

public string Name { get; set; }

public int CurrentRoom { get; set; }

public Player(string name, int currentRoom)

{

Name = name;

CurrentRoom = currentRoom;

}

}

[Serializable]

public class Room

{

public string Description { get; set; }

public List<Choice> Choices { get; set; }

public bool IsSolved { get; set; }

// Оставлен только первоначальный конструктор

public Room(string description)

{

Description = description;

Choices = new List<Choice>();

IsSolved = false;

}

public void AddChoice(string description, int nextRoom)

{

Choices.Add(new Choice(description, nextRoom));

}

}

[Serializable]

public class Game

{

public Player Player { get; set; }

public Room[] Rooms { get; set; }

public Game(Player player, Room[] rooms)

{

Player = player;

Rooms = rooms;

}

}

[TestMethod]

public void ChoiceConstructor\_InitializedCorrectly()

{

// Arrange

string description = "Test Description";

int nextRoom = 42;

// Act

Choice choice = new Choice(description, nextRoom);

// Assert

Assert.AreEqual(description, choice.Description);

Assert.AreEqual(nextRoom, choice.NextRoom);

}

[TestMethod]

public void PlayerConstructor\_InitializedCorrectly()

{

// Arrange

string playerName = "TestPlayer";

int currentRoom = 1;

// Act

Player player = new Player(playerName, currentRoom);

// Assert

Assert.AreEqual(playerName, player.Name);

Assert.AreEqual(currentRoom, player.CurrentRoom);

}

[TestMethod]

public void GameConstructor\_InitializedCorrectly()

{

// Arrange

Player player = new Player("TestPlayer", 1);

Room[] rooms = new Room[] { new Room("TestRoom") };

// Act

Game game = new Game(player, rooms);

// Assert

Assert.AreEqual(player, game.Player);

CollectionAssert.AreEqual(rooms, game.Rooms);

}

[TestMethod]

public void RoomConstructor\_InitializedCorrectly()

{

// Arrange

string roomDescription = "Test Room Description";

// Act

Room room = new Room(roomDescription);

// Assert

Assert.AreEqual(roomDescription, room.Description);

Assert.IsNotNull(room.Choices);

Assert.IsFalse(room.IsSolved);

Assert.AreEqual(0, room.Choices.Count);

}

[TestMethod]

private void AssertEqualGames(Game expected, Game actual)

{

Assert.AreEqual(expected.Player.Name, actual.Player.Name);

Assert.AreEqual(expected.Player.CurrentRoom, actual.Player.CurrentRoom);

Assert.AreEqual(expected.Rooms.Length, actual.Rooms.Length);

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

{

Assert.AreEqual(expected.Rooms[i].Description, actual.Rooms[i].Description);

Assert.AreEqual(expected.Rooms[i].IsSolved, actual.Rooms[i].IsSolved);

// Сравниваем Choices поэлементно

Assert.AreEqual(expected.Rooms[i].Choices.Count, actual.Rooms[i].Choices.Count);

for (int j = 0; j < expected.Rooms[i].Choices.Count; j++)

{

Assert.AreEqual(expected.Rooms[i].Choices[j].Description, actual.Rooms[i].Choices[j].Description);

Assert.AreEqual(expected.Rooms[i].Choices[j].NextRoom, actual.Rooms[i].Choices[j].NextRoom);

}

}

}

[TestMethod]

public void SaveAndLoadGame\_ShouldWorkCorrectly()

{

// Arrange

Player player = new Player("TestPlayer", 1);

Room[] rooms = {new Room("Test Room 1"), new Room("Test Room 2")};

rooms[0].AddChoice("Test Choice 1", 2);

rooms[1].AddChoice("Test Choice 2", 1);

Game originalGame = new Game(player, rooms);

// Act

SaveAndLoadGame(originalGame, out Game loadedGame);

// Assert

AssertEqualGames(originalGame, loadedGame);

}

private void SaveAndLoadGame(Game originalGame, out Game loadedGame)

{

// Save

SaveGame(originalGame);

// Load

LoadGame(out loadedGame);

}

private void SaveGame(Game game)

{

try

{

using (FileStream fs = new FileStream("savegame\_test.dat", FileMode.Create))

{

IFormatter formatter = new BinaryFormatter();

formatter.Serialize(fs, game);

}

}

catch (SerializationException ex)

{

Assert.Fail("Serialization failed: " + ex.Message);

}

catch (Exception ex)

{

Assert.Fail("Failed to save the game: " + ex.Message);

}

}

private void LoadGame(out Game game)

{

game = null;

try

{

using (FileStream fs = new FileStream("savegame\_test.dat", FileMode.Open))

{

IFormatter formatter = new BinaryFormatter();

game = (Game)formatter.Deserialize(fs);

}

}

catch (SerializationException ex)

{

Assert.Fail("Deserialization failed: " + ex.Message);

}

catch (Exception ex)

{

Assert.Fail("Failed to load the game: " + ex.Message);

}

}

}

}