МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

|  |
| --- |
| КАФЕДРА №42 |

КУРСОВАЯ РАБОТА (ПРОЕКТ)   
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | С.Ю. Гуков |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К РАБОТЕ ПО УЧЕБНОЙ ПРАКТИКЕ |
| Разработка игры |
| по дисциплине: |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | Z3423K | |  |  |  | М.А. Сергеенков |
|  | номер группы | |  | подпись, дата |  | инициалы, фамилия |
| Студенческий билет № | |  | |  |  |  |

|  |  |
| --- | --- |
| Шифр ИНДО |  |

Санкт-Петербург 2024

**Индивидуальное задание**

Разработать консольную игру, без графического интерфейса. Необходимо, реализовать в игре взаимодействие нескольких игроков и взаимодействие игрока с компьютером. Игра должна быть функциональной и иметь текстовый интерфейс для комфортной игры пользователем.

**Содержание**

[Введение 4](#_Toc168245617)

[1. Тема и описание игры 5](#_Toc168245618)

[2. Пользовательская документация 6](#_Toc168245619)

[2.1 Игра против человека 6](#_Toc168245620)

[2.1 Игра против компьютера 7](#_Toc168245621)

[Техническая документация 9](#_Toc168245622)

[Тестирование программы 10](#_Toc168245623)

[Заключение 11](#_Toc168245624)

[Список использованных источников 12](#_Toc168245625)

[Приложение с исходным кодом 13](#_Toc168245626)

# Введение

В настоящий период манеж компьютерных игр весьма известен в обществе. Каждый год появляются новейшие технологические процессы, которые ориентируют формирование новейших возможностей в сфере компьютерных игр. Компьютерные игры относятся к количеству важных вопросов современной науки, таким образом предполагают собою особенный продукт развития техники и современного человека. Игры могут помочь нам создать различные актуальные условия, проблемы и дают кое-какие возможные дороги для их решения. Игра содержит внутри себя все важные принципы для развития отдельно взятого человека и культуры сообщества. При улучшении компьютеров, улучшались и компьютерные игры, притягивая все большое количество людей. В настоящий промежуток компьютерное оборудование добилось такого степени развития, то что дает возможность создателям разрабатывать игры очень реалистичными, с отличной графикой и хорошим голосовым сопровождением и вследствие этого игры становятся всё более популярны для отдыха и развлечений.

В целом, видеоигры могут предложить множество преимуществ, помимо развлечений, включая когнитивное развитие, социальные связи, здравоохранение и карьерные возможности.

Разработка игр стала популярным выбором профессии для многих людей, которые увлечены видеоиграми и любят программирование. В наше время существует множество инструментов и программ, которые могут помочь и облегчить создание игр. Нужно посмотреть на все инструменты и выбрать, тот который больше всего подходит.

Цель учебной практики является закрепление и углубление теоретических знаний, полученных при изучении естественно-научных и профессиональных дисциплин; приобретение практических навыков и компетенций в сфере профессиональной деятельности; приобретение опыта применения современной вычислительной техники для решения практических задач.

# Тема и описание игры

Темой учебной практики «Разработка консольной игры». Я выбрал игру под название «Крестики-нолики». Логическая игра между двумя противниками или между игроком и компьютером на квадратном поле 3 на 3 клетки или полем больших размеров. Один игрок играет «крестиками», другой игрок использует «нолики». Игра начинается на пустом поле, задача каждого из игроков поместить свои фигуры в ряд горизонтально, вертикально или по диагонали. Игроки ходят по очереди, побеждает тот, кому первому удастся разместить правильно фигуры. Если на поле не остается пустых клеток, то победитель не выявлен и это считается ничья.

# Пользовательская документация

При запуске приложения пользователю нужно выбрать режим игры (Рисунок 1). В данной игре два режима. Первый режим «игра с человеком», в нем пользователь может играть против своих друзей, коллег и других людей. Второй режим «Игра с компьютером» в ним пользователь играет против компьютера.

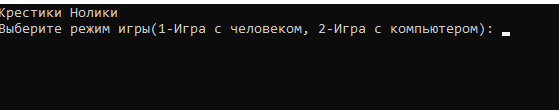


Рисунок 1 - Выбор режима игры

## **2.1 Игра против человека**

После выбора первого режима игрокам открывается условный интерфейс. В котором первый игрок должен выбрать координаты ячейки (Рисунок 2).

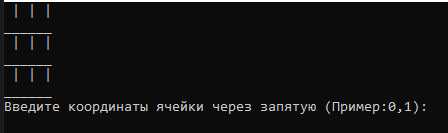


Рисунок 2 – Ввод координат ячейки

Потом в зависимости чей ход, игрок должен ввести нужный свой символ «Х» или «О» (Рисунок 3).

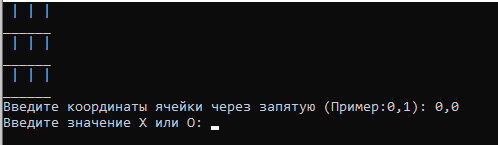


Рисунок 3 - Ввод своего символа

Затем он отобразить на игровом поле, и процедура повторить с вводом координат и символов (Рисунок 4).

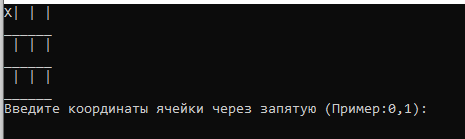


Рисунок 4 - Изменение игрового поля

После завершения игры игрокам выводиться надпись кто из них победил (Рисунок 5).

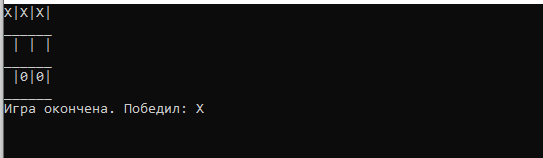


Рисунок 5 - Текстовое сообщение о победе игрока

## **2.1 Игра против компьютера**

В игре против компьютера будет все так же, только не придется выбирать символ. Он по умолчанию «Х», у компьютера соответственно «О» (Рисунок 6).

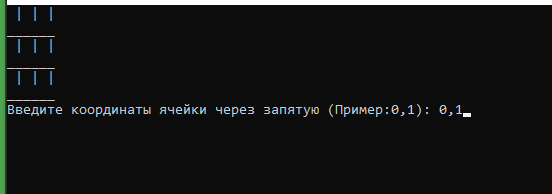


Рисунок 6 - Игровой процесс с компьютером

Игрок выбирает координаты ячейки и там появляется его символ, а в оставшийся, пустых полях, появляется символ компьютера (Рисунок 7).

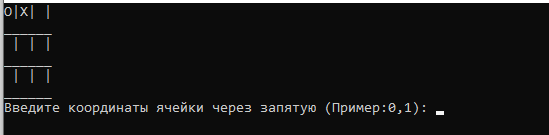


Рисунок 7 - Игровой процесс с компьютером

После окончание игры, так же выводиться победитель в строке (Рисунок 8)

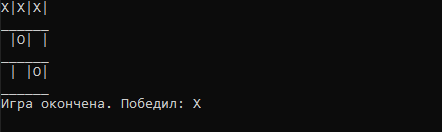


Рисунок 8 - Вывод победителя

# Техническая документация

В данной таблице описаны методы и их типы, которые использовались при создании игры (Таблица 1)

Таблица 1 – Методы игры

|  |  |
| --- | --- |
| Модификатор и тип | Метод и описание |
| void | GameOfBot Игра, где второй игрок компьютер |
| int | BotStep Метод возвращает случайное число для хода компьютера |
| bool, string | Checkwinner  Метод, который проверяет все поля на победные и возвращает есть ли победная и кто ее собрал |
| void | PrintPlace  Метод, который рисует игровое поле |
| void | GameOfPlayer  Игра, где второй игрок человек |

На данном рисунке изображена блок-схема функции Main. Служит отправной точкой программы (Рисунок 9).

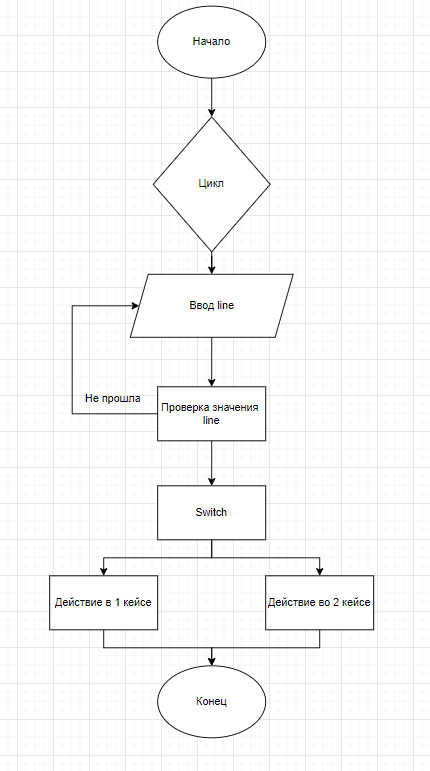


Рисунок 9 - Блок-схема Main

На данном рисунке изображена блок-схема метода PrintPlace.Она нужна для отрисовки игрового поля. (Рисунок 10).

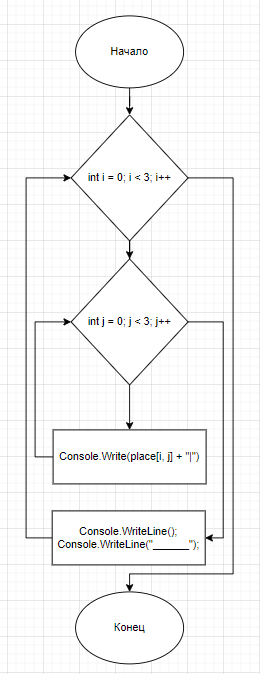


Рисунок 10 - Блок-схема PrintPlace

# Тестирование программы

Заключительным этапом разработки является тестирование и отладка продукта.

Отладка – это работа, направленная на обнаружение и исправление ошибок в программе.

Тестирование – это процесс выполнения программы на некотором наборе данных, для которого заранее известен результат или известны правила поведения этих программ.

# Тестирование мобильного приложения

Произведем тестирование программного продукта с применением технологии тест-кейсов. Результаты тест-кейсов отображены в таблицах 2, 3, 4 и 5.

Таблица 2 – Тест-кейс успешного запуска игры

|  |  |
| --- | --- |
| Название | Описание |
| Наименование проекта | Игровое приложение «Крестики-Нолики» |
| Номер версии проекта | 1.0 |
| Имя тестировщика | Максим |
| Дата тестирования | 22.05.2024 14:25 |
| Test Case # | TC\_2 |
| Приоритет тестирования | Высокий. |
| Название тестирования | Запуск игры. |
| Резюме испытания | Проверка корректности работы приложения при запуске игры. |
| Шаги тестирования | Запустить среду разработки.  Запустить приложение «Крестики-Нолики» в тестовом режиме. |
| Данные тестирования | Консольная игра |
| Ожидаемый результат | Запуск прошел успешно |
| Фактический результат | Запуск прошел успешно |
| Предпосылки | Запуск среды разработки  Запуск приложения «Крестики-Нолики» |
| Постусловия | Отсутствуют. |
| Статус *(Pass/Fail)* | Pass. |
| Комментарий | Никаких дополнительных требований не предусмотрено. |

Таблица 3 – Тест-кейс выбор режима игры

|  |  |
| --- | --- |
| Название | Описание |
| Наименование проекта | Игровое приложение «Крестики-Нолики» |
| Номер версии проекта | 1.0 |
| Имя тестировщика | Максим |
| Дата тестирования | 22.05.2024 14:35 |
| Test Case # | TC\_2 |
| Приоритет тестирования | Высокий. |
| Название тестирования | Успешный выбор режима |
| Резюме испытания | Проверка на корректный выбор режима |
| Шаги тестирования | 1. Запустить среду разработки. 2. Запустить приложение «Крестики-Нолики» в тестовом режиме. 3. Выбрать режим игры 4. Убедиться, что все работает корректно |
| Данные тестирования | Консольная игра |
| Ожидаемый результат | Режим запуститься корректно, без ошибок с работающим  функционалом |
| Фактический результат | Режим запуститься корректно, без ошибок с работающим  функционалом |
| Предпосылки | Запуск среды разработки  Запуск приложения «Крестики-Нолики» |
| Постусловия | Отсутствуют. |
| Статус *(Pass/Fail)* | Pass. |
| Комментарий | Никаких дополнительных требований не предусмотрено. |

Таблица 4 – Тест-кейс ввод координат ячейки

|  |  |
| --- | --- |
| Название | Описание |
| Наименование проекта | Игровое приложение «Крестики-Нолики» |
| Номер версии проекта | 1.0 |
| Имя тестировщика | Максим |
| Дата тестирования | 22.05.2024 14:45 |
| Test Case # | TC\_2 |
| Приоритет тестирования | Высокий. |
| Название тестирования | Правильная работа функции с вводом координат |
| Резюме испытания | Проверка на корректную работу ввода координат |
| Шаги тестирования | 1. Запустить среду разработки. 2. Запустить приложение «Крестики-Нолики» в тестовом режиме. 3. Выбрать режим игры 4. Ввести координаты ячейки 5. Убедиться, что все работает корректно |
| Данные тестирования | Консольная игра |
| Ожидаемый результат | Координаты успешно введены |
| Фактический результат | Координаты успешно введены |
| Предпосылки | Запуск среды разработки  Запуск приложения «Крестики-Нолики» |
| Постусловия | Отсутствуют. |
| Статус *(Pass/Fail)* | Pass. |
| Комментарий | Никаких дополнительных требований не предусмотрено. |

Таблица 5 – Тест-кейс корректной работы вывода победителя

|  |  |
| --- | --- |
| Название | Описание |
| Наименование проекта | Игровое приложение «Крестики-Нолики» |
| Номер версии проекта | 1.0 |
| Имя тестировщика | Максим |
| Дата тестирования | 22.05.2024 14:55 |
| Test Case # | TC\_2 |
| Приоритет тестирования | Высокий. |
| Название тестирования | Корректное отображение победителя |
| Резюме испытания | Проверка корректной работы вывода победителя |
| Шаги тестирования | 1. Запустить среду разработки. 2. Запустить приложение «Крестики-Нолики» в тестовом режиме. 3. Выбрать режим игры 4. Ввести координаты ячейки 5. Выбрать X или 0 6. Повторить 4 и 5 пункт до выигранной комбинации 7. Убедиться, что все работает корректно |
| Данные тестирования | Консольная игра |
| Ожидаемый результат | Победитель успешно введен |
| Фактический результат | Победитель успешно введен |
| Предпосылки | Запуск среды разработки  Запуск приложения «Крестики-Нолики» |
| Постусловия | Отсутствуют. |
| Статус *(Pass/Fail)* | Pass. |
| Комментарий | Никаких дополнительных требований не предусмотрено. |

Ссылка на короткое видео с демонстрацией механики игры;

[https://drive.google.com/file/d/1wlXampkDu69XZ\_Aj5gQSqfaM5B8U85i0/view?usp=sharing](https://drive.google.com/file/d/1wlXampkDu69XZ_Aj5gQSqfaM5B8U85i0/view?usp=sharing%20)

# Заключение

В данном проекте было разработано игровое приложение «Крестики-нолики». В ходе выполнения работы были освоены основные навыки разработки консольных приложений, закреплены базовые знания, полученные, в ходе изучения материала

Использовать созданное приложение, может любой человек для своего времяпрепровождения.

Данный проект может послужить отличным примером любому человеку, научиться думать совсем иначе, чем раньше.

Программа также может оказаться полезной в семейном кругу, с друзьями, коллегами. Так как в ней можно весело провести время, решить спор, либо устроить соревнование самого тактичного игрока.

Созданный в ходе учебной практики программный продукт обладает всеми необходимым для реализации поставленной в начале разработки задачи функциями:

* + Разработано консольное приложение, без графического интерфейса;
  + Реализовано возможность игры с человеком, а также с компьютером;
  + Реализована система победителя;
  + Выполнен упрощённый текстовый интерфейс.

# Список использованных источников

1. ГОСТ 7.32-2017 «СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления».
2. ГОСТ 2.105-2019 «ЕСКД. Общие требования к текстовым документам»;
3. ГОСТ Р 7.0.100-2018 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления».
4. https://habr.com – русскоязычное IT сообщество.
5. https://tproger.ru/ – публикации для разработчиков
6. https://stackoverflow.com – заграничный ресурс обмена опытом среди разработчиков.
7. https://stepik.org/catalog - курсы для программирования.
8. Краткое введение в Git и его использование с GitHub - GitHub - ashtanyuk/Git-intro: Краткое введение в Git и его использование с GitHub, https://github.com/ashtanyuk/Gitintro (дата посещения: 06.02.2023)

# Приложение с исходным кодом

//Основной блок с кодом игры

static void Main()

{

//Создание двумерного массива

string[,] gamePlace = new string[3, 3] { { " ", " ", " " }, { " ", " ", " " }, { " ", " ", " " } };

while (true)

{

Console.Clear();

Console.WriteLine("Крестики Нолики");

Console.Write("Выберите режим игры(1-Игра с человеком, 2-Игра с компьютером): ");

string line = Console.ReadLine();

int result;

int.TryParse(line, out result);

// Кейсы, открывающиеся в зависимости что выбрал игрок

switch (result)

{

case 1:

gamePlace = new string[3, 3] { { " ", " ", " " }, { " ", " ", " " }, { " ", " ", " " } };

GameOfPlayer(gamePlace);

break;

case 2:

gamePlace = new string[3, 3] { { " ", " ", " " }, { " ", " ", " " }, { " ", " ", " " } };

GameOfBot(gamePlace);

break;

}

}

}

//Игра с компьютером

private static void GameOfBot(string[,] gamePlace)

{

bool isGameOver = false;

bool isBot = false;

while (!isGameOver)

{

int a = 0, b = 0;

string line;

Console.Clear();

PrintPlace(gamePlace);

//Вывод результата

var result = Checkwinner(gamePlace);

if (result.Item1 && result.Item2 != " ")

{

Console.WriteLine($"Игра окончена. Победил: {result.Item2}");

Console.ReadLine();

isGameOver = true;

continue;

}

//Ввод данных и проверка их

if (!isBot)

{

Console.Write("Введите координаты ячейки через запятую (Пример:0,1): ");

line = Console.ReadLine();

bool parseA = false;

if (line != "")

{

parseA = char.IsNumber(line[0]);

}

if (parseA == false)

{

Console.WriteLine("Первой ячейки не существует, введите существующее значение");

Console.ReadLine();

continue;

}

else if (a < 0 || a > 2)

{

Console.WriteLine("Первой ячейки не существует, введите существующее значение");

Console.ReadLine();

continue;

}

else

{

a = Convert.ToInt16(line[0].ToString());

}

bool isNext = false;

for (int i = 1; i < line.Length; i++)

{

if (line[i].ToString() == ",")

{

isNext = true;

}

if (isNext && line[i].ToString() != " ")

{

bool parseB = char.IsNumber(line[i]);

if (parseB)

{

b = Convert.ToInt16(line[i].ToString());

break;

}

}

}

if (b < 0 || b > 2)

{

Console.WriteLine("Второй ячейки не существует, введите существующее значение");

Console.ReadLine();

continue;

}

}

if (isBot)

line = "O";

else

line = "X";

if (gamePlace[a, b] == " " && line != " " && isBot == false)

{

gamePlace[a, b] = line;

isBot = true;

}

else if (isBot)

{

while (true)

{

int x, y;

x = BotStep();

y = BotStep();

if (gamePlace[x, y] == " ")

{

gamePlace[x, y] = line;

isBot = false;

break;

}

}

}

else

{

Console.WriteLine("Это поле занято или вы ничего не ввели");

Console.ReadLine();

continue;

}

}

Console.ReadLine();

}

//Генерация случаного числа для хода компьютера

private static int BotStep()

{

Random bot = new Random();

return bot.Next(0, 3);

}

//Метод проверки побеждающей комбинации

private static (bool, string) Checkwinner(string[,] place)

{

//по строке

if ((place[0, 0] == place[0, 1]) && (place[0, 1] == place[0, 2]) && place[0, 0] != " ")

return (true, place[0, 0]);

if ((place[1, 0] == place[1, 1]) && (place[1, 1] == place[1, 2]) && place[1, 0] != " ")

return (true, place[1, 0]);

if ((place[2, 0] == place[2, 1]) && (place[2, 1] == place[2, 2]) && place[2, 0] != " ")

return (true, place[2, 0]);

// по столбцу

if ((place[0, 0] == place[1, 0]) && (place[1, 0] == place[2, 0]) && place[0, 0] != " ")

return (true, place[0, 0]);

if ((place[0, 1] == place[1, 1]) && (place[1, 1] == place[2, 1]) && place[0, 1] != " ")

return (true, place[0, 1]);

if ((place[0, 2] == place[1, 2]) && (place[1, 2] == place[2, 2]) && place[0, 2] != " ")

return (true, place[0, 2]);

// по диагонали

if ((place[0, 0] == place[1, 1]) && (place[1, 1] == place[2, 2]) && place[0, 0] != " ")

return (true, place[0, 0]);

if ((place[0, 2] == place[1, 1]) && (place[1, 1] == place[2, 0]) && place[0, 2] != " ")

return (true, place[0, 2]);

return (false, " ");

}

//Отрисовка игрового поля

private static void PrintPlace(string[,] place)

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

Console.Write(place[i, j] + "|");

}

Console.WriteLine();

Console.WriteLine("\_\_\_\_\_\_");

}

}

//Игра со вторым игроком

private static void GameOfPlayer(string[,] gamePlace)

{

bool isGameOver = false;

while (!isGameOver)

{

int a = 0, b = 0;

string line;

Console.Clear();

PrintPlace(gamePlace);

//Вывод результата

var result = Checkwinner(gamePlace);

if (result.Item1 && result.Item2 != " ")

{

Console.WriteLine($"Игра окончена. Победил: {result.Item2}");

Console.ReadLine();

isGameOver = true;

continue;

}

//Ввод данных и проверка их

Console.Write("Введите координаты ячейки через запятую (Пример:0,1): ");

line = Console.ReadLine();

bool parseA = false;

if (line != "")

{

parseA = char.IsNumber(line[0]);

}

if (parseA == false)

{

Console.WriteLine("Первой ячейки не существует, введите существующее значение");

Console.ReadLine();

continue;

}

else if (a < 0 || a > 2)

{

Console.WriteLine("Первой ячейки не существует, введите существующее значение");

Console.ReadLine();

continue;

}

else

{

a = Convert.ToInt16(line[0].ToString());

}

bool isNext = false;

for (int i = 1; i < line.Length; i++)

{

if (line[i].ToString() == ",")

{

isNext = true;

}

if (isNext && line[i].ToString() != " ")

{

bool parseB = char.IsNumber(line[i]);

if (parseB)

{

b = Convert.ToInt16(line[i].ToString());

break;

}

}

}

if (b < 0 || b > 2)

{

Console.WriteLine("Второй ячейки не существует, введите существующее значение");

Console.ReadLine();

continue;

}

Console.Write("Введите значение X или O: ");

line = Console.ReadLine();

line = line.ToUpper();

if (gamePlace[a, b] == " " && line != " ")

{

gamePlace[a, b] = line;

}

else

{

Console.WriteLine("Это поле занято или вы ничего не ввели");

Console.ReadLine();

continue;

}

}

Console.ReadLine();

}