Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно-графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Алгоритмизация и программирование*** |
|  |  |
| на тему | Разработка программы |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020-РГР-02.03.02-№ 22-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Семикин Николай Дмитриевич | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | ФИТ-**232** | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | *Фундаментальная информатика и информационные технологии* | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | |  | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2023

Содержание

[1. ВВЕДЕНИЕ 3](#_Toc155364782)

[2. ТЕОРИЯ 4](#_Toc155364783)

[3. РАЗРАБОТКА И КОД ПРОГРАММЫ 5](#_Toc155364784)

[4. РЕЗУЛЬТАТЫ 9](#_Toc155364785)

[5. ЗАКЛЮЧЕНИЕ 10](#_Toc155364786)

[6. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 11](#_Toc155364787)

**1. ВВЕДЕНИЕ**

Сегодня информационные технологии и программирование применяются во всех сферах жизни общества. Начальным уровнем разработки в программировании является создание консольного приложение, именно этот аспект будет рассмотрен в данной расчетно-графической работе.

В данной работе будет рассмотрена разработка консольного приложения, которое представляет собой программу, выполняемую в текстовом интерфейсе командной строки. Консольные приложения являются важной частью программирования, поскольку они позволяют взаимодействовать с пользователем через команды и текстовый ввод-вывод.

Целью данной работы является изучение основных принципов разработки консольных приложений, а также их применение на практике. В ходе работы будет рассмотрено создание консольного приложения на примере определенной задачи, а также принципы работы с командной строкой, обработка ввода-вывода, управление процессами и другие аспекты, связанные с консольным программированием.

**2. ТЕОРИЯ**

Разработка консольного приложения в рамках данной работы будет проводится на языке C#. Я выбрал следующую задачу для выполнения: играющий выбирает одну из трех лошадей, состязающихся в бегах, и выигрывает, если его лошадь приходит первой, скорость передвижения лошадей на разных этапах выбирается программой с помощью датчика случайных чисел («Ипподром»).

Для выполнения выбранной задачи мне понадобятся знания о вводе и выводе информации с использованием консоли, умение составлять алгоритмы для обработки введенной информации, а также информация о том, как создавать случайные числа.

**3. РАЗРАБОТКА И КОД ПРОГРАММЫ**

В основе программы лежит следующая схема: игрок получает список лошадей, из которых он может выбрать одну, далее каждой лошади с помощью генератора случайных чисел присваивается скорость от одного до тысячи, выбирается лошадь с максимальной скоростью, ее номер будет считаться выигрышным, далее, введенный игроком номер сравнивается с выигрышным, если они совпадут, то игрок выиграл, в противном случае – проиграл, информация об этом будет выведена, после чего игроку будет предложено сыграть еще раз. В случае, если скорости лошадей совпадут, то выигравшей будет считаться лошадь с наименьшим номером.

Во время разработки программы я решил включить в ее алгоритм проверку вводимых пользователем данных, чтобы код не выдавал ошибку, даже если были введены неправильные данные, например, несуществующий или отрицательный номер лошади или текст в поле, принимающее числовое значение.

Программа имеет следующую структуру. Основной алгоритм, выполняется в цикле, пока игрок не захочет закончить.

Первый блок программы содержит вывод имен доступных лощадей:

Console.WriteLine("Horses:");

Console.WriteLine(" " + "1. Olivia");

Console.WriteLine(" " + "2. Max");

Console.WriteLine(" " + "3. Antony");

Второй блок программы осуществляет ввод и проверку номера, выбранного игроком:

int n = 0;

while (n != 1 & n != 2 & n != 3)

{

Console.Write("Which one do you prefer? ");

string try\_n = Console.ReadLine();

bool check\_n = true;

foreach (char value in try\_n)

{

bool digit = char.IsDigit(value);

if (digit==false) { check\_n = false; }

}

if(check\_n == true)

{

n = Convert.ToInt32(try\_n);

}

}

Третий блок программы содержит генерацию случайной скорости лошадей и выбор номера лошади с максимальной скоростью:

Random rnd = new Random();

int[] speed = { rnd.Next(1, 1000), rnd.Next(1, 1000), rnd.Next(1, 1000) };

int max\_speed = Math.Max(Math.Max(speed[0], speed[1]), speed[2]);

int win = 0;

for(int i = 0; i < 3; i++) { if (speed[i] == max\_speed) { win = i + 1; break; } }

Для написания данного блока я воспользовался интернет-ресурсом Tutorialsteacher.com для получения информации о модуле Random.

Четвертый блок программы осуществляет вывод информации о победившей лошади:

Console.WriteLine($"Horse number {win} have won!");

if (n == win)

{

Console.WriteLine("Congratulations! You've won :)");

}

else

{

Console.WriteLine("Sorry, you've lost :(");

}

Пятый блок программы запрашивает у игрока, желает ли тот продолжить, если да, то цикл запускается заново, если нет, то программа выходит из цикла и консольное окно закрывается:

string c = " ";

while(c != "Y" & c != "N")

{

Console.Write("Write Y if you want to play again, N if you don't: ");

c = Console.ReadLine();

}

if(c == "N"){ break; }

Весь код программы представлен ниже:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using static System.Net.Mime.MediaTypeNames;

namespace Ippodrom

{

internal class Program

{

static void Main()

{

while(true)

{

Console.WriteLine("Horses:");

Console.WriteLine(" " + "1. Olivia");

Console.WriteLine(" " + "2. Max");

Console.WriteLine(" " + "3. Antony");

int n = 0;

while (n != 1 & n != 2 & n != 3)

{

Console.Write("Which one do you prefer? ");

string try\_n = Console.ReadLine();

bool check\_n = true;

foreach (char value in try\_n)

{

bool digit = char.IsDigit(value);

if (digit==false) { check\_n = false; }

}

if(check\_n == true)

{

n = Convert.ToInt32(try\_n);

}

}

Random rnd = new Random();

int[] speed = { rnd.Next(1, 1000), rnd.Next(1, 1000), rnd.Next(1, 1000) };

int max\_speed = Math.Max(Math.Max(speed[0], speed[1]), speed[2]);

int win = 0;

for(int i = 0; i < 3; i++) { if (speed[i] == max\_speed) { win = i + 1; break; } }

Console.WriteLine($"Horse number {win} have won");

if (n == win)

{

Console.WriteLine("Congratulations! You've won :)");

}

else

{

Console.WriteLine("Sorry, you've lost :(");

}

string c = " ";

while(c != "Y" & c != "N")

{

Console.Write("Write Y if you want to play again, N if you don't: ");

c = Console.ReadLine();

}

if(c == "N"){ break; }

}

}

}

}

**4. РЕЗУЛЬТАТЫ**

Результат работы программы представлен на рисунке 1.

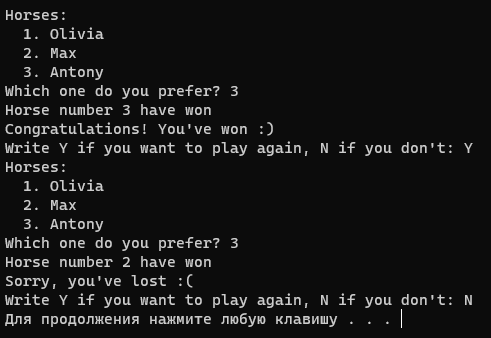


Рисунок – 1

На рисунке 2 показана работа программы в случае, если игрок попробует ввести неподходящие данные.

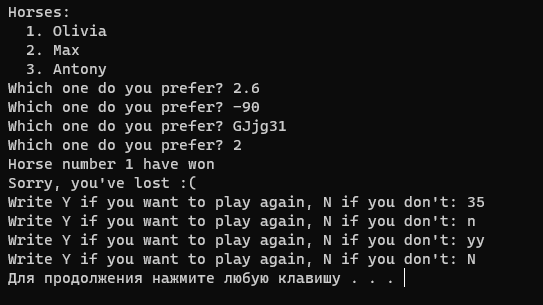


Рисунок – 2

**5. ЗАКЛЮЧЕНИЕ**

В ходе выполнения расчетно-графической работы я создал консольное приложение, нацеленное на решение специфической поставленной передо мной задачи. Я работал с вводом, выводом данных и их обработкой, а также изучил получение случайных значений на языке C#.

Созданное приложение выполняет поставленную задачу и предусматривает случаи введения пользователем некорректных данных.

**6. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1) Tutorialsteacher.com (05.01.24): <https://www.tutorialsteacher.com/articles/generate-random-numbers-in-csharp>