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-№ 14-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Колодницкого Ильи Михайловича | | | | | |
|  |  |  |  | | 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 2024

Оглавление

[Введение 2](#_Toc155968376)

[Теоретическая часть 2](#_Toc155968377)

[Разработка кода 2](#_Toc155968378)

[Результаты 2](#_Toc155968379)

[Заключение 2](#_Toc155968380)

[Список использованной литературы 2](#_Toc155968381)

# Введение

Программирование является одним из ключевых процессов, направленных на разработку эффективных программ и алгоритмов, которые могут выполняться на вычислительных устройствах. С развитием электронных устройств и их широкого использования, появилось множество различных средств программирования. Наша повседневная жизнь существенно изменилась благодаря использованию различных гаджетов и технологий, которые стали неотъемлемой частью нашей современной эры.

В современных информационных технологиях программирование играет важную роль, значительно улучшая нашу жизнь по многим аспектам. Оно способствует удобству, эффективности и информативности нашего повседневного существования. Программирование продолжает преобразовывать мир, предлагая новые возможности в сферах образования, бизнеса и развлечений. Это неотъемлемая часть инновационных технологий, которая сильно влияет на разные стороны нашей жизни. Мы ведь видим результаты программирования в мобильных приложениях, интернет-сервисах и социальных сетях.

Профессионалы в области программирования сейчас очень востребованы, так как они создают и поддерживают технологические решения для сложных задач. Программирование позволяет нам автоматизировать повседневные процессы, что экономит время и ресурсы. Благодаря появлению новых языков программирования и фреймворков, разработка программных продуктов стала более быстрой и эффективной. В настоящее время программирование не только способствует техническому прогрессу, но и помогает решать социальные и экологические проблемы. Обучение программированию становится важной частью образовательной программы, открывая двери для новых поколений в цифровом мире.

# Теоретическая часть

Тема работы заключается в разработке алгоритма на языке С# для игры "Угадай число". Созданный алгоритм будет случайным образом выбирать число из заданного диапазона и предоставлять игроку возможность угадать его. Игрок может указать свою попытку, и если она совпадает с загаданным числом, выводится сообщение об успехе. В противном случае игрок получает подсказку, что его число слишком мало или слишком велико. Игра продолжается до тех пор, пока игрок не угадает число или не исчерпает все попытки.

В С# каждый класс содержит методы, которые выполняют определенные действия или вычисляют значения. Методы объединяют операторы, которые выполняются вместе. Все методы принадлежат классу и для их вызова необходимо использовать имя метода и определить параметры, если они есть. Если метод находится в другом классе, то он должен быть объявлен как public и можно вызвать его по имени класса и метода.

В C# есть операторы выбора if и switch, которые основываются на выполняющемся условии и выбирают соответствующие операторы для выполнения. Операторы цикла, такие как while, do - while, for и foreach, выполняют операции до тех пор, пока условие остается верным.

Программа в С# состоит из последовательности операторов, которые выполняются последовательно. Чтобы группировать операторы, используются блоки с фигурными скобками. В каждом блоке определены локальные переменные, которые существуют только в его области видимости.

В ООП ключевым словом является класс. При запуске приложения в С# начинает выполняться метод Main, который является точкой входа. В приложении может быть много классов, но запускается только один метод Main, который выбирается при компиляции. При завершении метода Main приложение заканчивает свою работу. Пространства имен используются для организации классов и подключения сторонних библиотек с помощью ключевого слова using.

Комментарии в С# используются для объяснения кода и улучшения его читаемости. Они могут использоваться для отладки кода и предотвращения выполнения определенных участков кода. Виды комментариев - однострочные и многострочные, которые помогают описывать классы, методы, свойства, поля, параметры и переменные. Хорошо написанные комментарии должны быть краткими и информативными, описывать действия кода, а не его реализацию, и не дублировать код, а дополнять его.

# Разработка кода

Данный код представляет собой консольное приложение на языке программирования C#, которое реализует игру "Угадай число".

Изначально генерируется случайное число N с помощью объекта класса Random. Затем, с помощью этого же объекта, генерируется случайное число random\_number в диапазоне от 0 до N.

После этого выводится сообщение с информацией о том, что программа загадала случайное число от 0 до N и предлагает пользователю угадать его.

Далее генерируется случайное число attempt, которое означает количество попыток у пользователя для угадывания числа. Это число выводится на экран.

Затем в цикле проходит процесс угадывания числа. Пользователь вводит свое предполагаемое число, сравнивается с загаданным числом. Если числа совпадают, то выводится сообщение о победе и количество попыток, за которое удалось угадать число. Если число меньше загаданного, выводится сообщение "Слишком мало. Возьмите побольше.". Если число больше загаданного, выводится сообщение "Слишком много. Возьмите поменьше.".

Если у пользователя заканчиваются попытки, то выводится сообщение о поражении и предлагается попытаться еще раз. После этого пользователю дается еще неограниченное количество попыток для угадывания числа до тех пор, пока он не угадает число.

Таким образом, данный код реализует игру "Угадай число", где пользователь должен угадать случайное число, которое генерируется программой в определенном диапазоне, используя ограниченное количество попыток.

using System;

class HelloWorld

{

static void Main()

{

Random number = new Random();

// берет рандомный диапазон от 0 до 100000

int N = number.Next(0,100001);

// число, которое нужно угадать

int random\_number = number.Next(N+1);

Console.WriteLine($"Программа загадала случайное число от 0 до {N}. Попробуй угадать)");

// рандомное кол-во попыток от 10 до 35

int attempt = number.Next(10,36);

Console.WriteLine();

Console.WriteLine($"Даю тебе {attempt} попыток");

Console.WriteLine();

//кол-во попыток

int cnt = 1;

//вводим переменную для того, чтобы в дальнейшем ее использовать

int your\_number = -1;

//угадываем с теми попытками которые даст программа

for (int i = attempt; i > 0; i--, cnt++)

{

Console.WriteLine("-----------------------------------");

Console.WriteLine($"У тебя осталось {i} попыток");

//вводим свое число

your\_number = Convert.ToInt32(Console.ReadLine());

if (your\_number == random\_number)

{

Console.WriteLine();

Console.WriteLine($"Ура!!! Вы смогли угадать число за {cnt} попыток");

break;

}

else if (your\_number < random\_number)

{

Console.WriteLine("Слишком мало. Возьмите побольше.");

}

else

{

Console.WriteLine("Слишком много. Возьмите поменьше.");

}

//если попытки закончились выводим сообщение

if (i - 1 == 0)

{

Console.WriteLine();

Console.WriteLine($"Увы, вы не смогли угадать число за {attempt} попыток. Но ничего страшного, пытайтесь еще!!! Вы справитесь!!!!!!!!");

cnt--;

}

}

//продолжаем угадывать если попытки закончились

while (your\_number != random\_number)

{

Console.WriteLine("-----------------------------------");

//вводим свое число

your\_number = Convert.ToInt32(Console.ReadLine());

if (your\_number == random\_number)

{

Console.WriteLine();

Console.WriteLine($"Ура!!! Вы смогли угадать число за {cnt+1} попыток");

break;

}

else if (your\_number < random\_number)

{

Console.WriteLine("Слишком мало. Возьмите побольше.");

}

else

{

Console.WriteLine("Слишком много. Возьмите поменьше.");

}

cnt++;

Console.WriteLine($"Вы уже потратили {cnt} попыток");

}

}

}

# Результаты

На рисунке 1 представлен результат программы, в котором у пользователя закончились попытки для угадывания, но он все еще может продолжить угадывать. Также пользователю отображается сколько попыток он использовал.

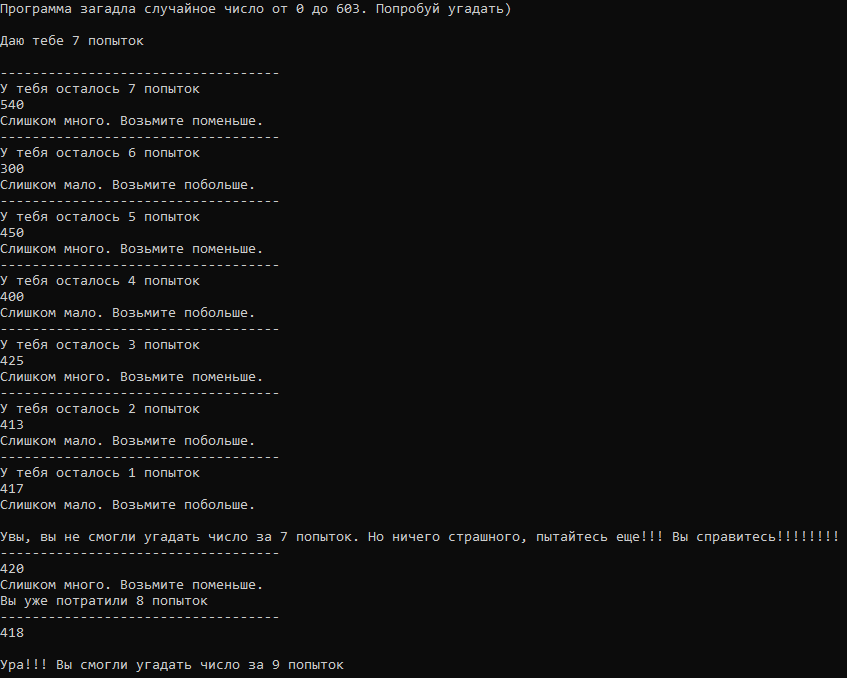


Рисунок 1 – Результат программы

На рисунке 2 представлен результат программы, в котором пользователь угадал число за данное количество попыток.

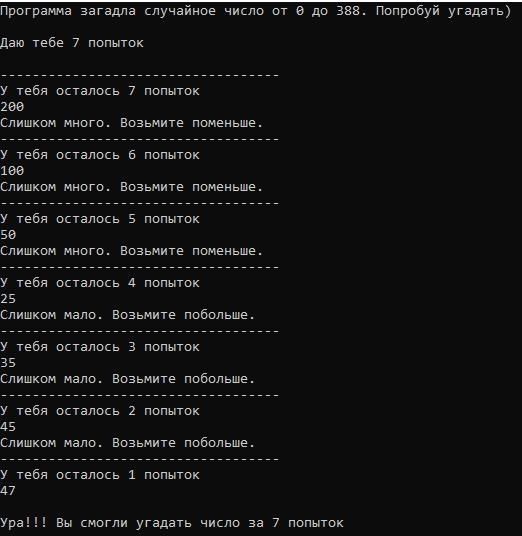


Рисунок 2 – Результат программы

На рисунке 3 представлен результат программы, в котором пользователь угадал число за количество попыток меньшее тому количеству которое дал компьютер.

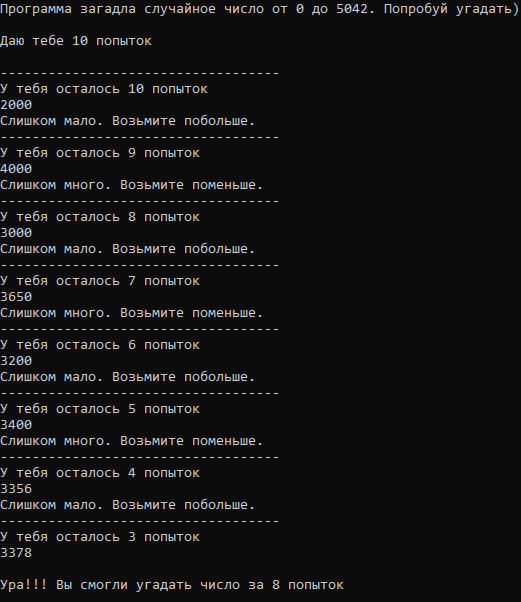


Рисунок 3 – Результат программы

# Заключение

В ходе выполнения данной расчетно-графической работы была разработана игра "Угадай число". Был создан класс "HelloWorld" с методом "Main", который является точкой входа в программу, и который реализует основную функциональность игры.

Основной функционал программы включает в себя генерацию случайного числа, ввод количества попыток и осуществление игры, где пользователь пытается угадать загаданное число. Результаты игры (количество попыток) выводятся на экран.

Программа написана на языке программирования C# и использует стандартные библиотеки. Она представляет собой пример простой консольной игры, демонстрирующей навыки работы с условиями, циклами и методами в C#.

# Список использованной литературы

# Генерация случайных чисел в C# - <https://csharp.webdelphi.ru/generaciya-sluchajnyx-chisel-v-c/> (Дата обращения: 10.01.24)

# Generating Random Numbers in C# - <https://devtut.github.io/csharp/generating-random-numbers-in-c.html#generate-a-random-int> (Дата обращения: 10.01.24)

# Random Класс - <https://learn.microsoft.com/ruru/dotnet/api/system.random?view=net-80> (Дата обращения: 10.01.24)