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-№ 23-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Тимаковой Лилии Максимовны | | | | | |
|  |  |  |  | | 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**

[Введение 3](#_Toc156211794)

[Теория 4](#_Toc156211795)

[Разработка кода программы 6](#_Toc156211796)

[Результаты 9](#_Toc156211797)

[Заключение 13](#_Toc156211798)

[Список литературы 14](#_Toc156211799)

# **Введение**

В современном мире очень широко используются информационные технологии. Они проникли во все сферы нашей жизни, от коммуникаций между людьми и развлечений до бизнеса и науки. Причины такого широкого использования информационных технологий -это их способность обеспечивать быстрый и удобный доступ к информации, автоматизировать процессы, упрощать коммуникацию и сотрудничество, а также анализировать данные, моделировать сложные процессы. В целом, широкое использование информационных технологий в современном мире улучшает качество жизни, повышает эффективность работы и способствует развитию науки.

# **Теория**

Суть игры в слова, моделируемой данной программой, - за определенное изначально количество попыток угадать буквы случайного слова, загаданного другим игроком, при этом за каждый ход можно назвать одну букву. Программная реализация этой игры основана на случайном выборе. Здесь программа должна выбрать случайное число в диапазоне индексов слов, чтобы было выбрано слово с случайным индексом.

Бывают ситуации, в частности данная, когда требуется, чтобы результат работы программы был случайным в определенных пределах.

В докомпьютерные времена случайные числа получали, вытаскивая разноцветные мячи из мешков, вытягивая карты, бросая кости. Понятно, что серьёзные исследования так проводить было нельзя, поэтому в 1927 года Типпетт опубликовал первую таблицу случайных чисел. Чуть позже люди попытались автоматизировать этот процесс. Начали появляться машины, генерирующие случайные числа. Сейчас такие устройства тоже используются и называются источниками (генераторами) энтропии. Стоит заметить, что только такие устройства могут давать по-настоящему случайные числа. Но, к сожалению, генераторы энтропии дороги, и не возможно установить их в каждый ПК. Именно поэтому и возникла необходимость в алгоритмах получения случайных чисел.

Для реализации такой возможности во многих языках программирования присутствуют встроенные функции, код которых выдает случайные числа. На самом деле числа не совсем случайные, а псевдослучайные. Дело в том, что искусственно реализовать случайность невозможно. В большинстве случаев псевдослучайных чисел хватает.

Суть в том, что оно получается путем вычислений. И тут мы сталкиваемся с тем, что генерация истинно случайных чисел, по определению не может быть никак вычисленной. Из-за того, что во многих подобных функциях используются алгоритмы(чаще всего похожие, либо модифицирован один и тот же), они не могут считаться истинно случайными. Обычно берется некоторый коэффициент и с его помощью вычисляется каждое последующее "случайное" число.

Как мы понимаем, для того что бы получить истинно случайное число, у нас не должно быть каких либо вычислений, не считая входные данные, которые не всегда обязательны. Для этого нужно выйти за рамки компьютера, нам нужны какие-нибудь данные или средства из реального мира, к тому-же, что то хаотичное.

Запаздывающие генераторы Фибоначчи используются в классе Random на платформе .NET(C#). Это генераторы псевдослучайных чисел, также называемые аддитивными генераторами. Однако это лишь метод, который является основой для алгоритмов.

Аддитивные генераторы генерируют вместо случайных битов, случайные слова.

Также для программной реализации игры в слова потребуются средства консольного ввода- вывода, циклы, условные алгоритмы.

# **Разработка кода программы**

На вход в программу подаются буквы слова, которые вводит пользователь. они обрабатываются с точки зрения присутствия в составе загаданного слова. на выход подается счет играющего на каждой итерации цикла пока и слово с частью отгаданных букв, Оповещение о проигрыше\ победе.

Функция ReadWordsFromFile открывает файл и считывает из него данные в массив с элементами строкового типа, возвращает массив.

Функция PrintArrSpace принимает символьный массив и добавляет после каждого элемента массива пробел и выводит на экран массив.

В функции Main() выводятся сообщения для пользователя. Вызывается функция ReadWordsFromFile. Создается объект для генерации случайных чисел в диапазоне индексов элементов массива. В переменную s помещается элемент массива со сгенерированном случайным индексом. В переменную score помещается начальное значение счета игрока. Обьявляется массив с символами “\_”, которые будут заменяться на угаданные буквы и выводиться вместе с сообщением о начале игры. В цикле while у пользователя есть счет и массив с угаданными буквами, выполняется цикл пока угаданное не совпадает с загаданным словом. Предлагается ввести пользователю букву (не пробел и не другой символ).

В цикле for проверяется наличие введенной буквы в загаданном слове, если найдена, то добавляется на место “\_”,нет - декремент score.

Осуществляется печать результата хода и счета на каждой итерации.

Если по выходу из цикла score остался неотрицательным и слово отгадано- поздравление ,победа , если же счет 0 – сообщение, что игра окончена ,слово выводится.

**Код программы**

using System;

using System.Linq.Expressions;

class Wordplay

{

static string[] ReadWordsFromFile(string filename)

{

string[] words = File.ReadAllLines(filename);

return words;

}

static void PrintArrSpace(char[] array)

{

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

{

Console.Write(array[i] + " ");

}

}

static void Main()

{

Console.WriteLine("Игра в слова. Вы угадываете слово, загаданное программой.");

Console.WriteLine("Ваш счет 15. Если Вы угадаете букву, то она добавится на место \_ в слове. Иначе счет уменьшается на 1. Вы выигрываете, если угадаете слово, пока счет не стал равен 0");

string[] words = ReadWordsFromFile("words.txt");

Random random = new Random();

string s = words[random.Next(words.Length)];

int score = 15;

char[] Letters = new char[s.Length];

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

{

Letters[i] = '\_';

}

Console.WriteLine("В слове {0} букв. У Вас счет {1}.", s.Length, score);

PrintArrSpace(Letters);

while (score > 0 && new string(Letters) != s)

{

Console.Write("Введите букву: ");

char guess = Console.ReadLine()[0];

if (Char.IsWhiteSpace(guess) || !Char.IsLetter(guess))

{

Console.WriteLine("Вы ввели не букву или пробел!");

break;

}

bool found = false;

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

{

if (s[i] == guess)

{

Letters[i] = guess;

found = true;

}

}

if (!found)

{

score--;

}

PrintArrSpace(Letters);

Console.WriteLine("Ваш счет: {0}", score);

Console.WriteLine();

}

if (score >= 0 && new string(Letters) == s)

{

Console.WriteLine("Поздравляем! Вы отгадали слово");

}

if (score == 0)

{

Console.WriteLine("Игра окончена! Было загадано слово: {0}", s);

}

Console.ReadLine();

}

}

# **Результаты**

На рисунках 1-2 представлены результаты работы программы в случае не отгадывания слова, на рисунке 3-в случае отгадывания, 4-если неправильно введен символ.

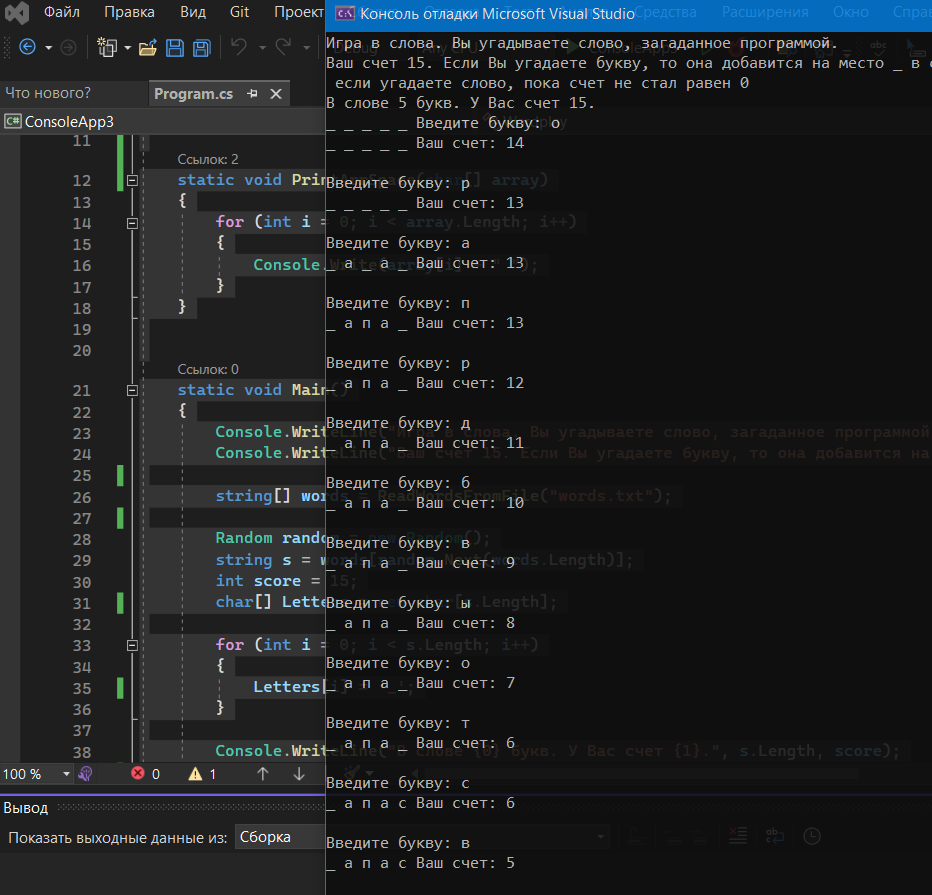


Рисунок 1 – Результат работы с не отгаданным словом(начало)

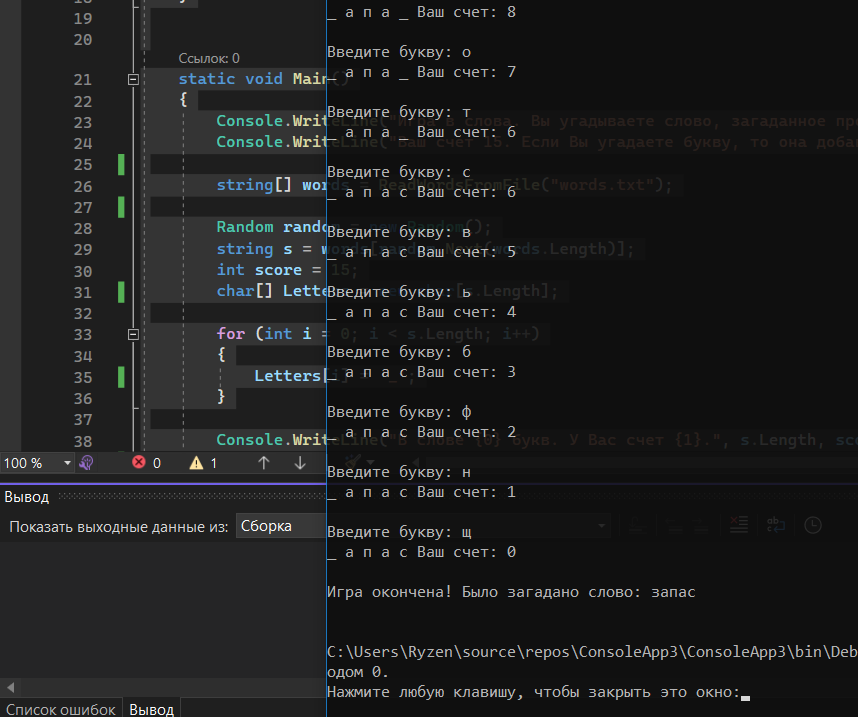


Рисунок 2 – Результат работы с не отгаданным словом(конец)



Рисунок 3 – Результат работы с отгаданным словом

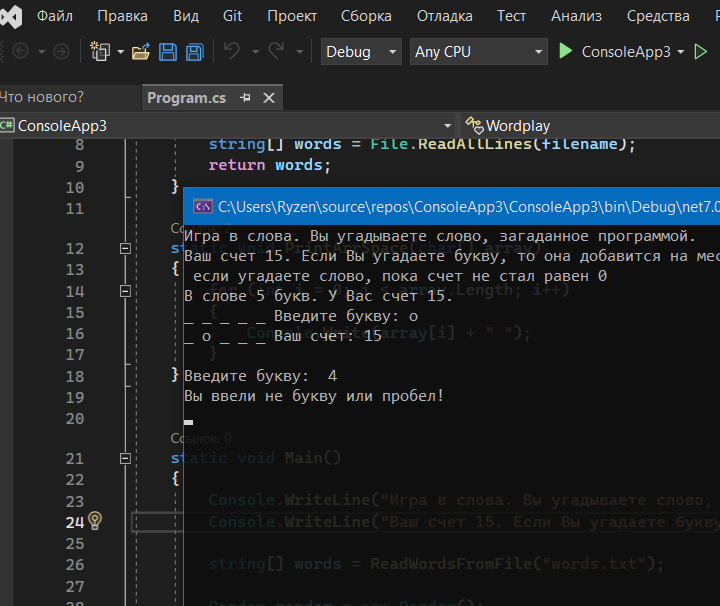


Рисунок 4 – Результат работы с неправильным символом

# **Заключение**

В ходе выполнения РГР была разработана программа для моделирования игры в слова с компьютером, изучен способ генерации случайных чисел в C#, использованы алгоритмические конструкции, ветвление и цикл.

# **Список литературы**

1. habr.com.ru (сайт) URL:https://habr.com/ru/articles/686190(14.01.23)
2. pas1.ru (сайт) URL:https://pas1.ru/random(14.01.23)
3. habr.com.ru (сайт) URL:https://habr.com/ru/articles/132217(14.01.23)
4. proglib.io(сайт) URL:https://proglib.io/p/how-computers-make-random-numbers(14.01.23)
5. plssite.ru(сайт)URL:http://plssite.ru/csharp/csharp\_random\_article.html(14.01.23)