МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский авиационный институт (национальный исследовательский университет)»

Ступинский филиал МАИ

Кафедра «Моделирование систем и информационные технологии»

Лабораторная работа № 3

**Тема ЛР: ,,Написание программы для подсчета факториала числа и отображения порядкового элемента из ряда Фибоначчи на c#”**

Студент: Ивинский И. А.

Группа: ТСО-105Б-22

Дисциплина: «Информатика»

Преподаватель: Новиков Б.Б.

Ступино 2023

**Цель работы:** Изучение языка программирования c# путем закрепления изученной теории на практике.

**Задачи:** написать программу, которая:

* Считает факториал введенного пользователем числа и выводит на консоль ответ;
* Выводит на консоль число из ряда Фибоначчи по введенному пользователем порядковому номеру;
* Cодержит меню для выбора пользователем, какие подсчеты делать;
* Осуществляет проверку на ввод числа пользователем, чтобы он не вводил буквы и символы.

**Описание выбранных технологий:**

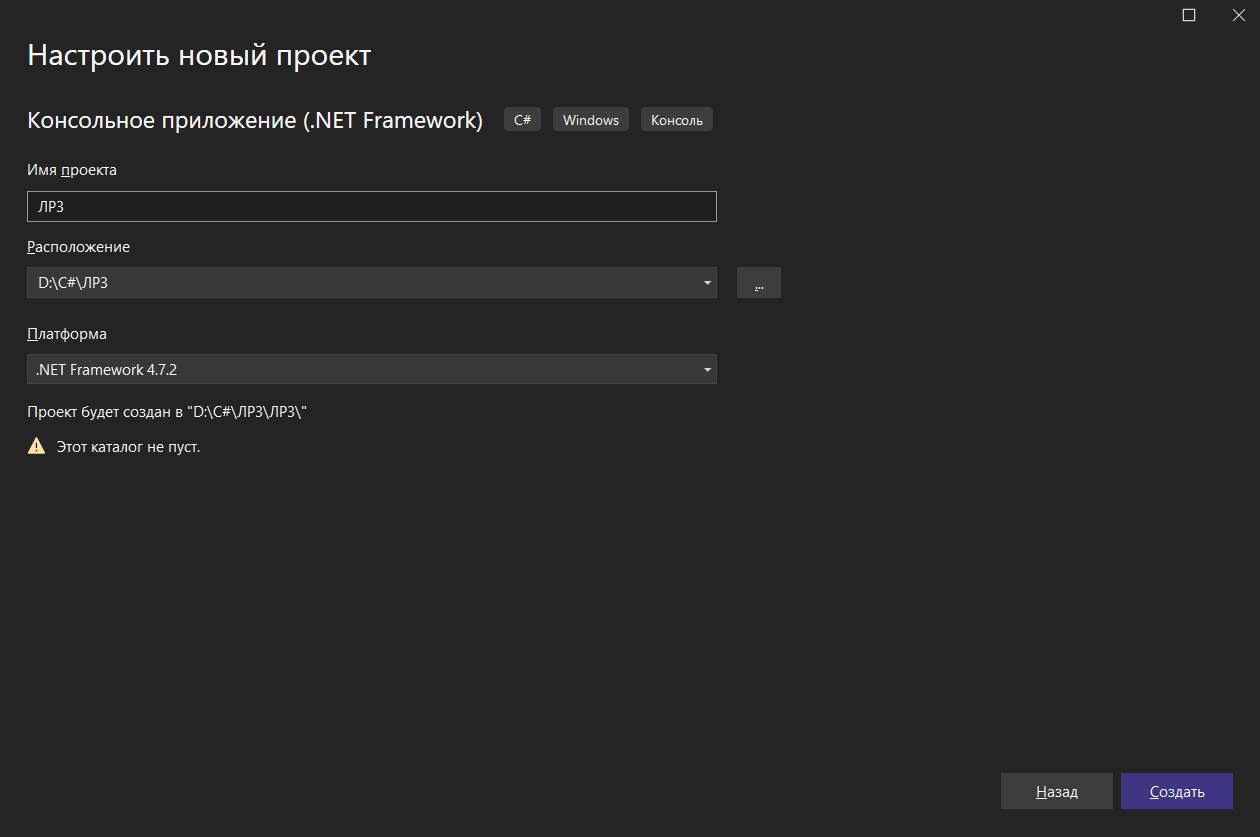
1. C# — объектно-ориентированный, ориентированный на компоненты язык программирования, позволяющий разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET.

2. Visual studio - приложение для редактирования кода.

3. Вспомогательный ресурс - learn.microsoft.com.

**Ход работы:**

1. Создание проекта или среды для разработки текстового квеста:

****

2. Процесс написания кода, принцип работы инструкций:

Одной из инструкций является класс Console и его метод WriteLine(): Console.WriteLine(), куда внесен текст, отображаемый в консоли и написанный в двойных кавычках "".

Внутри метода WriteLine(), помимо текстовых литералов (констант) используется управляющая последовательность "\n", функция которой - перенос на новую, следующую строку после строки, на которую перемещает текст метод WriteLine(). Используется для улучшения читабельности и работает через бинарный оператор +:

Console.WriteLine("\n"+"текст"+"\n")

Также для данного класса будет использоваться метод ReadLine инструкции Console.ReadLine(). Функция которого заключается в cчитывании строки символов из стандартного входного потока. Эту строку можно присвоить объявленной переменной с типом данных строка:

string wish = Console.ReadLine();

**Для проверки** на ввод данных пользователем, чтобы он не вводил буквы и символы, используется бесконечный цикл while, в круглых скобках после while прописывается условие, после чего в фигурных задается действие, которое необходимо совершать, пока условие не будет выполнено:

while (условие))

{

действие

}

while (!int.TryParse(Console.ReadLine(), out x))

{

Console.WriteLine("\n" + "Вы ввели некорректные данные, повторите ввод: " + "\n");

}

**В данном случае необходимо пробовать преобразовать вводимую пользователем строку в число целочисленного типа. Пока это не будет получаться (ввиду того, что пользователь вводит не только цифры), на экран консоли будет выводиться сообщение о неправильно введенных данных.**

**Человек может ввести число, факториал от которого превышает доступные значения, записывающиеся в переменную, поэтому через конструкции** if**,** else if**,** else**, задаются ограничения, при прохождении через которые пользователь будет получать сообщение о том, что программа или законы математики (факториал отрицательного числа) не позволят осуществить операцию с такими числами:**

if (x > 12)

{

Console.WriteLine("\n" + "Переменная не может принять такое большое значение и отобразить на экране, введите число меньше 13");

}

else if (x < 0)

{

Console.WriteLine("\n" + "Факториала отрицательного числа не существует");

}

else

{

int y = 1;

while (x > 1)

{

y \*= x;

x--;

}

Console.WriteLine("\n" + $"Факториал введенного числа равен {y}");

}

Код, позволяющий считать факториал от введенного пользователем числа:

int y = 1;

while (x > 1)

{

y \*= x;

x--;

}

Переменной x уже присвоено значение, введенное пользователем. Объявим переменную y и присвоим значение 1. В цикле while, пока выполняется условие, мы будем перемножать вновь полученный y с уменьшенным x и переприсваивать значение. x, уменьшаясь на единицу, выполняет функцию порядково убывающего числа, как это делается в математике: 5!=5\*4\*3\*2\*1. По невыполнении условий мы получим окончательный y с нужным значением.

Далее объявляется переменная со значением true, благодаря циклу while можем задать инструкцию, что пока функция MainMenu() возвращает true, пользователь может взаимодействовать с новым консольным окном, написанным внутри метода MainMenu():

bool showMenu = true;

while (showMenu)

{

showMenu = MainMenu();

}

Далее само создание метода:

private static bool MainMenu()

private **- модификатор доступа, означающий, что доступ** к закрытым членам можно получить только внутри тела класса или структуры, в которой они объявлены.

Модификатор static используется для объявления статического члена, принадлежащего собственно типу, а не конкретному объекту.

bool означает, что метод будет возвращать булевое значение.

В теле метода прописывается код, очищающий консоль от предыдущих операций, создается меню для выбора пользователем, какие подсчеты делать:

Console.Clear();

Console.WriteLine("Значение какого по порядку элемента из ряда Фибоначчи вы бы хотели узнать?");

Console.WriteLine("0)");

Console.WriteLine("17)");

Console.WriteLine("39)");

Console.Write("\r\nВведите порядковый номер: ");

Для рассмотрения конкретных, выбранных пользователем, вариантов используется оператор switch case. Его принцип работы заключается в том, что происходит выполнение одного выражения case из множества возможных путей на основе значения этого выражения. Как пример, если переменная, взятая в switch, может принимать минимум два значения, то каждый case которому присвоено одно из этих значений, будет выполнять инструкции только при равенстве переменной этому значению.

int g;

while (!int.TryParse(Console.ReadLine(), out g))

{

Console.WriteLine("\n" + "Вы ввели некорректные данные, повторите ввод: " + "\n");

}

switch (g)

{

case 0:

Console.Clear();

Console.WriteLine("Первый элемент ряда равен 0");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

return true;

case 17:

Console.Clear();

Console.WriteLine("17 элемент ряда равен 1597");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

return true;

case 39:

Console.Clear();

Console.WriteLine("39 элемент ряда равен 63245986");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

return true;

Инструкция case по окончании своего действия возвращает значение true, так как метод имеет тип возвращаемого значения bool.

**Затем расписывается случай, выбранный пользователем и отличный от предложенных. В данной ситуации уже будет вестись подсчет самой программой - какое число стоит под номером, указанным пользователем:**

default:

if (g>48)

{

Console.Clear();

Console.WriteLine($"Число {g} слишком велико и не охватывается типом данных, введите число до 48");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

}

else if (g<0)

{

Console.Clear();

Console.WriteLine("Программа рассматривает ряд Фибоначчи из положительных чисел и нуля");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

}

else {

Console.Clear();

int first = 1;

int second = 1;

int sum = 0;

int j = 2;

while (j <= g)

{

sum = first + second;

first = second;

second = sum;

j++;

}

Console.WriteLine($"Под номером {g} в ряде Фибоначчи стоит число {first}");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

}

return true;

**Код целиком:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите число, факториал которого вы бы хотели узнать: " + "\n");

int x;

while (!int.TryParse(Console.ReadLine(), out x))

{

Console.WriteLine("\n" + "Вы ввели некорректные данные, повторите ввод: " + "\n");

}

if (x > 12)

{

Console.WriteLine("\n" + "Переменная не может принять такое большое значение и отобразить на экране, введите число меньше 13");

}

else if (x < 0)

{

Console.WriteLine("\n" + "Факториала отрицательного числа не существует");

}

else

{

int y = 1;

while (x > 1)

{

y \*= x;

x--;

}

Console.WriteLine("\n" + $"Факториал введенного числа равен {y}");

}

Console.WriteLine("\n" + "Введите число, факториал которого вы бы хотели узнать: " + "\n");

int b;

while (!int.TryParse(Console.ReadLine(), out b))

{

Console.WriteLine("\n" + "Вы ввели некорректные данные, повторите ввод: " + "\n");

}

if (b > 12)

{

Console.WriteLine("\n" + "Переменная не может принять такое большое значение и отобразить на экране, введите число меньше 13");

}

else if (b < 0)

{

Console.WriteLine("\n" + "Факториала отрицательного числа не существует");

}

else { int fac = 1;

for (int l = b; l > 1; l--)

{

fac \*= l;

if (l == 2)

{

Console.WriteLine("\n" + $"Факториал введенного числа равен {fac}");

}

}

}

Console.WriteLine("\n" + "Введите число, факториал которого вы бы хотели узнать: " + "\n");

int z;

while (!int.TryParse(Console.ReadLine(), out z))

{

Console.WriteLine("\n" + "Вы ввели некорректные данные, повторите ввод: " + "\n");

}

if (z > 12)

{

Console.WriteLine("\n" + "Переменная не может принять такое большое значение и отобразить на экране, введите число меньше 13");

}

else if (z < 0)

{

Console.WriteLine("\n" + "Факториала отрицательного числа не существует");

}

else { int v = 1;

do

{

v \*= z;

z--;

}

while (z > 1);

Console.WriteLine("\n" + $"Факториал введенного числа равен {v}");

Console.WriteLine("\n" + "Нажмите enter, чтобы перейти к ряду Фибоначчи");

Console.ReadLine();

}

bool showMenu = true;

while (showMenu)

{

showMenu = MainMenu();

}

}

private static bool MainMenu()

{

Console.Clear();

Console.WriteLine("Значение какого по порядку элемента из ряда Фибоначчи вы бы хотели узнать?");

Console.WriteLine("0)");

Console.WriteLine("17)");

Console.WriteLine("39)");

Console.Write("\r\nВведите порядковый номер: ");

int g;

while (!int.TryParse(Console.ReadLine(), out g))

{

Console.WriteLine("\n" + "Вы ввели некорректные данные, повторите ввод: " + "\n");

}

switch (g)

{

case 0:

Console.Clear();

Console.WriteLine("Первый элемент ряда равен 0");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

return true;

case 17:

Console.Clear();

Console.WriteLine("17 элемент ряда равен 1597");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

return true;

case 39:

Console.Clear();

Console.WriteLine("39 элемент ряда равен 63245986");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

return true;

default:

if (g>48)

{

Console.Clear();

Console.WriteLine($"Число {g} слишком велико и не охватывается типом данных, введите число до 48");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

}

else if (g<0)

{

Console.Clear();

Console.WriteLine("Программа рассматривает ряд Фибоначчи из положительных чисел и нуля");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

}

else {

Console.Clear();

int first = 1;

int second = 1;

int sum = 0;

int j = 2;

while (j <= g)

{

sum = first + second;

first = second;

second = sum;

j++;

}

Console.WriteLine($"Под номером {g} в ряде Фибоначчи стоит число {first}");

Console.Write("\r\nНажмите enter, чтобы вернуться");

Console.ReadLine();

}

return true;

}

}

}

}

**Заключение: Написана работающая программа для подсчета факториала числа и отображения порядкового элемента из ряда Фибоначчи на c#.**