1. **ПРОЦЕДУРЫ И ФУНКЦИИ – МЕТОДЫ КЛАССА**

Задание 1. Разработать метод F(x), который в двузначном числе меняет цифры местами, а остальные числа оставляет без изменения.

Листинг программы:

using System.Text;

try

{

Console.Write("Введите число: ");

string number = Console.ReadLine()

number = ChangePlaces(number);

Console.WriteLine(number);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

string ChangePlaces(string number)

{

if (number.Length == 2)

{

StringBuilder sb = new StringBuilder(number);

char elem = sb[0];

sb[0] = sb[1];

sb[1] = elem;

number = sb.ToString();

}

return number;

}

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 21 | 12 |

Анализ результатов:

C:\Users\1234321\Downloads\govno1.png

Рисунок 3.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Построить таблицу значений функции.

Листинг программы:

try

{

Console.Write("Введите начало диапазона: ");

double start = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите конец диапазона: ");

double end = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите длину шага: ");

double step = Convert.ToDouble(Console.ReadLine())

for (double i = start; i <= end; i += step)

{

Console.WriteLine($"f({Math.Round(i, 1)}) = {Math.Round(result(i), 1)}");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

static double result (double x)

{

double y = 0;

if (x < 0)

y = 0;

else if (x == 1)

y = 1;

else if (x > 0 && x != 1)

{

y = Math.Pow(x, 2) + 1;

}

return y;

}

Таблица 3.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| -1,1,0.5 | 0,0,0,1.2,1 |

Анализ результатов:

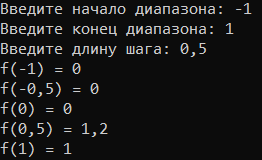


Рисунок 3.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Перегрузить метод F из предыдущего раздела так, чтобы его сигнатура соответствовала виду static void f (double x, out double y).

Листинг программы:

class Program

{

public static void Main(string[] args)

{

try

{

Console.Write("Введите начало диапазона: ");

double start = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите конец диапазона: ");

double end = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите длину шага: ");

double step = Convert.ToDouble(Console.ReadLine());

double y;

for (double i = start; i <= end; i += step)

{

Result(i, out y);

Console.WriteLine($"f({Math.Round(i, 1)}) = {Math.Round(y, 1)}");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

static double Result(double x)

{

double y = 0;

if (x < 0)

y = 0;

else if (x == 1)

y = 1;

else if (x > 0 && x != 1)

{

y = Math.Pow(x, 2) + 1;

}

return y;

}

static void Result(double x, out double y)

{

y = 0;

if (x < 0)

y = 0;

else if (x == 1)

y = 1;

else if (x > 0 && x != 1)

{

y = Math.Pow(x, 2) + 1;

}

}

}

static double Result(double x)

{

double y = 0;

if (x < 0)

y = 0;

else if (x == 1)

y = 1;

else if (x > 0 && x != 1)

{

y = Math.Pow(x, 2) + 1;

}

return y;

}

static void Result(double x, out double y)

{

y = 0;

if (x < 0)

y = 0;

else if (x == 1)

y = 1;

else if (x > 0 && x != 1)

{

y = Math.Pow(x, 2) + 1;

}

}

}

Таблица 3.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| -3,3,1.5 | 0,0,0,3.2,10 |

Анализ результатов:

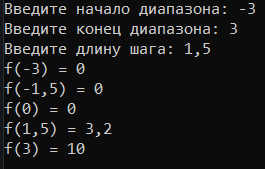


Рисунок 3.3 – Результат работы программы

Источник: собственная разработка

Задание 4. Описать класс для работы с двумерным массивом целых чисел.

Реализовать возможность нахождения числа, полученного перемножением

положительных элементов массива, меньших 10.

Листинг программы:

try

{

int rowCount = 5;

int colCount = 5;

Matrix mass = new Matrix(5, 5);

Console.WriteLine("Сгенерировать стандартную матрицу? (y/n)");

char choice = Convert.ToChar(Console.ReadLine());

if (choice == 'y')

{

mass.MatrixInit();

}

else if (choice == 'n')

{

Console.Write("Введите количество строк матрицы: ");

rowCount = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите количество столбцов матрицы: ");

colCount = Convert.ToInt32(Console.ReadLine());

mass = new Matrix(rowCount, colCount);

mass.MatrixInit(rowCount, colCount);

}

for (int i = 0; i < mass.matr.GetLength(0); i++)

{

for (int j = 0; j < mass.matr.GetLength(1); j++)

Console.Write($"{mass.matr[i, j]}\t");

Console.WriteLine();

}

Console.WriteLine($"Результат умножения положительных элементов массива больше 10 = {mass.MultLess10()}");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class Matrix

{

public int[,] matr { get; set; }

public int rCount;

public int cCount;

private int \_result = 1;

public Matrix(int rowCount, int colCount)

{

this.rCount = rowCount;

this.cCount = colCount;

}

Random rand = new Random();

public void MatrixInit()

{

matr = new int[rCount, cCount];

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

{

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

{

matr[i, j] = rand.Next(30);

}

}

}

public void MatrixInit(int rowCount, int colCount)

{

matr = new int[rowCount, colCount];

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

{

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

{

matr[i, j] = rand.Next(30);

}

}

}

public int MultLess10()

{

return matr.Cast<int>().Where(x => x > 0 && x < 10).Aggregate((a, b) => a \* b);

}

}

Анализ результатов:

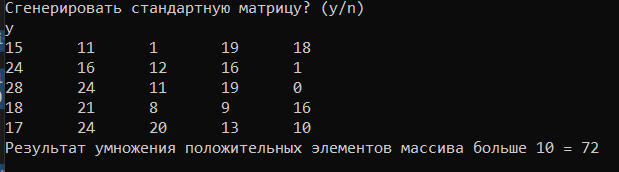


Рисунок 3.4 – Результат работы программы

Источник: собственная разработка

Задание 5. Описать класс «вокзал», содержащий закрытый массив поездов.

Обеспечить следующие возможности:

- вывод информации о поезде по номеру с помощью индекса;

- вывод информации о поездах, отправляющихся после введенного с

клавиатуры времени;

- перегруженную операцию сравнения, выполняющую сравнение

времени отправления двух поездов;

- вывод информации о поездах, отправляющихся в заданный пункт

назначения.

Информация должна быть отсортирована по времени отправления.

Написать программу, демонстрирующую все разработанные элементы

классов.

Листинг программы:

Train[] park = Station.FillStation();

Station station = new Station(park);

bool contin = true;

while (contin)

{

Console.WriteLine("Выеберите критерий:\n" +

"1. Получить информацию о поезде по номеру\n" +

"2. Получить информацию о поездах, отправляющихся после указанного времени\n" +

"3. Сравнение времени отправления 2 поездов\n" +

"4. Получить информацию о поездах, отправляющихся в заданный пункт назначения. \n" +

"5. Выход\n");

int choice = Int32.Parse(Console.ReadLine());

switch (choice)

{

case 1:

Console.WriteLine("Введите номер поезда: ");

int index = Convert.ToInt32(Console.ReadLine());

station.PrintTrainInfoByIndex(index);

break;

case 2:

Console.WriteLine("Введите час отправки: ");

int h = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите минуту отправки: ");

int m = Convert.ToInt32(Console.ReadLine());

station.PrintTrainInfoByTime(h, m);

break;

case 3:

Console.WriteLine("Введите номер первого поезда: ");

int first = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите номер первого поезда: ");

int second = Convert.ToInt32(Console.ReadLine());

station.CompareTime(first, second);

break;

case 4:

Console.WriteLine("Введите пункт назначения: ");

string destination = Console.ReadLine();

station.PrintTrainInfoByDestination(destination);

break;

case 5:

contin = false;

break;

}

}

public class Station

{

private Train[] \_train;

public Station(Train[] \_train)

{

this.\_train = \_train;

}

public static Train[] FillStation()

{

Train[] \_train = new Train[2];

Console.WriteLine("Заполните описание поездов");

for (int i = 0; i <= \_train.Length - 1; i++)

{

Console.WriteLine("Пункт назначения: ");

string dest = Console.ReadLine();

Console.WriteLine("Введите час отправки: ");

int timeH = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите минуту отправки: ");

int timeM = Convert.ToInt32(Console.ReadLine());

\_train[i] = new Train(i, dest, timeH, timeM);

}

return \_train;

}

public void PrintTrainInfoByIndex(int index)

{

for (int i = 0; i <= \_train.Length - 1; i++)

{

if (i == index)

\_train[i].Print();

}

}

public void PrintTrainInfoByTime(int h, int m)

{

for (int i = 0; i <= \_train.Length - 1; i++)

{

if (\_train[i].timeH > h (\_train[i].timeH == h && \_train[i].timeM > m))

\_train[i].Print();

}

}

public void PrintTrainInfoByDestination(string destination)

{

for (int i = 0; i <= \_train.Length - 1; i++)

{

if (\_train[i].destination == destination)

\_train[i].Print();

}

}

public void CompareTime(int first, int second)

{

Train fir = \_train[first];

Train sec = \_train[second];

if (fir > sec)

Console.WriteLine("Первый поезд отправляется позже");

else if (fir < sec)

Console.WriteLine("Второй поезд отправляется позже");

}

}

public class Train

{

public int number;

public string destination;

public int timeH;

public int timeM;

public Train(int number, string destination, int timeH, int timeM)

{

this.number = number;

this.destination = destination;

this.timeH = timeH;

this.timeM = timeM;

}

public void Print()

{

Console.WriteLine($"Номер поезда: {number}, пункт назначения: {destination}, время отправления: {timeH} ч, {timeM} м");

}

public static bool operator >(Train a, Train b)

{

if (a.timeH > b.timeH (a.timeH == b.timeH && a.timeM > b.timeM))

return true;

else

return false;

}

public static bool operator <(Train a, Train b)

{

if (a.timeH < b.timeH || (a.timeH == b.timeH && a.timeM < b.timeM))

return true;

else

return false;

}

}

Анализ результатов:

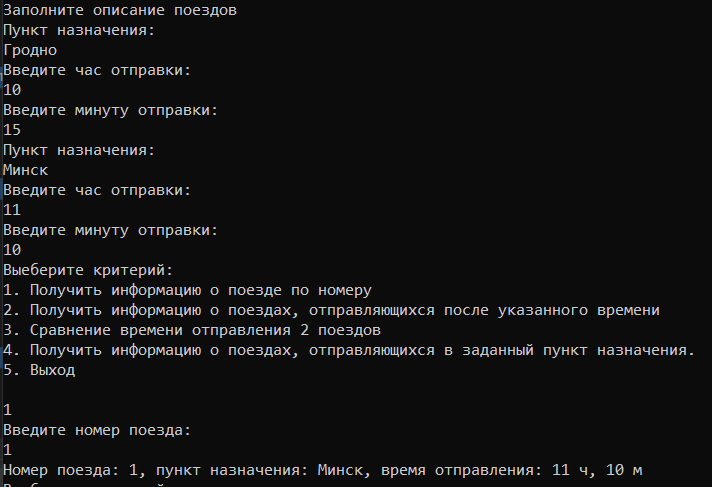


Рисунок 3.5 – Результат работы программы

Источник: собственная разработка