**Лабораторная работа №15\_1**

## Постановка задачи:

**Вариант 9**

1. Задать произвольный вещественный одномерный массив фиксированной длинны. Создать методы сортировки и обработки массива:

• Все числа, большие среднего по модулю – по убыванию, остальные – по возрастанию.

• Каждый предыдущий умножить на следующий и представить каждый полученный элемент в определенной валюты с соответствующим символом. Порядок элементов массива не изменяется.

С помощью делегатов, switch-case и try-catch предоставить пользователю выбор метода сортировки или обработки массива, выводя результат на экран. Предусмотреть случай, когда пользователь вводит не верные данные.

2. С клавиатуры вводится два некоторых числа. С помощью делегата сообщить пользователю их НОК, а также сообщить является одно из введенных чисел кубом какого-то числа. Предусмотреть случай, когда пользователь вводит не верные данные.

3. Реализовать метод, вычисляющий значение функции Cos(2\*x) и Tan(-5\*x) с заданным диапазоном (интервалом для вычислений)

[-25;25] с заданным шагом 0.4. Виды функций, передаваемых в качестве параметра оформить в виде делегата.

4. Задан класс Учебная группа, в котором есть поля количества девушек и юношей. Когда пользователь вводит с клавиатуры начальное количество девушек и юношей, он получает сообщение о событии, что создана такая группа. Через механизм switch-case ему можно добавить людей в группу или отчислить некоторое количество студентов, причем пользователь отчисляет любых студентов так, чтобы юношей и девушек должно быть примерно поровну, а добавлять может только юношей. Количество пользователь снова вводит с клавиатуры. Предусмотреть случай, когда пользователь вводит количество отчисленных больше, чем общее количество студентов, и случай, когда пользователь вводит не существующее значение в switch-case, а также возможные исключения try-catch.

5. Решить задачу номер 4 с использованием класса УчебнаяГруппаEventArgs.

## Код:

using System;

using System.Collections.Generic;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

namespace Test\_2

{

class StudyGroupEventArgs

{

public string Message { get; }

public int MaleCount { get; }

public int FemaleCount { get; }

public StudyGroupEventArgs(string msg, int mCount, int fCount)

{

Message = msg;

MaleCount = mCount;

FemaleCount = fCount;

}

public StudyGroupEventArgs(string msg)

{

Message = msg;

}

}

class StudyGroup

{

public delegate void StudyGroupHandler(object sender, StudyGroupEventArgs e);

public event StudyGroupHandler Created;

public event StudyGroupHandler Added;

public event StudyGroupHandler Deducted;

private int maleCount;

private int femaleCount;

public StudyGroup(int mCount, int fCount)

{

maleCount = mCount;

femaleCount = fCount;

}

public void Add(int mCount)

{

if (Math.Abs(maleCount + mCount - femaleCount) >= 5)

{

Console.WriteLine("Большая разница в количестве юношей и девушек!");

}

else

{

maleCount += mCount;

Added?.Invoke(this, new StudyGroupEventArgs($"Ещё зачислено {mCount} юношей"));

}

}

public void Deduct(int mCount, int fCount)

{

if ((mCount <= maleCount) && (fCount <= femaleCount))

{

if (Math.Abs((maleCount - mCount) - (femaleCount - fCount)) >= 5)

{

Console.WriteLine("Большая разница в количестве юношей и девушек!");

}

else

{

maleCount -= mCount;

femaleCount -= fCount;

Deducted?.Invoke(this, new StudyGroupEventArgs($"Отчислено {mCount} юношей и {fCount} девушек"));

}

}

else if (mCount > maleCount || fCount > femaleCount)

{

Console.WriteLine("Слишком мало людей в группе");

}

}

public void Information()

{

Console.WriteLine($"\nЮношей: {maleCount} \nДевушек: {femaleCount} \n");

Created?.Invoke(this, new StudyGroupEventArgs($"На курс зачислено {maleCount} юношей и {femaleCount} девушек", maleCount, femaleCount));

}

}

class MainClass

{

public delegate void MessageFirst(float[] array);

public delegate void MessageSecond(int a, int b);

public delegate double MessageThird(double x);

public static void Main(string[] args)

{

Console.OutputEncoding = Encoding.Unicode;

First();

Second();

Third();

Fourth();

}

public static void First()

{

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

Console.WriteLine("Задание 1");

MessageFirst pr = SortFirst;

float[] array = new float[10] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

int choice = 0;

Console.WriteLine("\n1. Сортировка по заданию \n2. Обработка по заданию \n3. Выход");

while (choice != 3)

{

try

{

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

switch (choice)

{

case 1: pr = SortFirst; break;

case 2: pr = ArrayProcessingFirst; break;

case 3: break;

default: Console.WriteLine("Некорректные данные!"); break;

}

if (choice == 3)

{

break;

}

pr(array);

}

catch (Exception ex)

{

Console.WriteLine("Некорректные данные!" + ex.GetBaseException());

}

Console.WriteLine();

}

Console.ReadLine();

}

public static void Second()

{

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

Console.WriteLine("Задание 2");

MessageSecond ms = null;

bool correctness = false;

while (!correctness)

{

try

{

Console.Write("a = "); int a = Convert.ToInt32(Console.ReadLine());

Console.Write("b = "); int b = Convert.ToInt32(Console.ReadLine());

ms += NOKSecond;

ms += CubeCheckingSecond;

ms(a, b);

correctness = true;

}

catch

{

Console.WriteLine("Некорректные данные!");

correctness = false;

}

}

Console.ReadLine();

}

public static void Third()

{

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

Console.WriteLine("Задание 3");

try

{

TrigonomCalcThird(Math.Cos, Math.Tan, -25, 25);

}

catch

{

Console.WriteLine("Некорректные данные!");

}

Console.ReadLine();

}

public static void Fourth()

{

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

Console.WriteLine("Задание 4");

bool correctness = false;

int choice = 0, mCount = 0, fCount = 0;

Console.WriteLine("Создай группу студентов");

while (!correctness)

{

try

{

Console.Write("Юношей: "); mCount = Convert.ToInt32(Console.ReadLine());

Console.Write("Девушек: "); fCount = Convert.ToInt32(Console.ReadLine());

if (Math.Abs(mCount - fCount) >= 5)

{

Console.WriteLine("Большая разница в количестве юношей и девушек!");

correctness = false;

}

else

{

correctness = true;

}

}

catch

{

Console.WriteLine("Некорректные данные!");

correctness = false;

}

}

StudyGroup studyGroup = new StudyGroup(mCount, fCount);

studyGroup.Added += ShowMsg;

studyGroup.Created += ShowMsg;

studyGroup.Deducted += ShowMsg;

studyGroup.Information();

studyGroup.Created -= ShowMsg;

Console.WriteLine("\n1. Зачислить студентов(только юношу) \n2. Отчислить студентов \n3. Показать информацию по группе \n4. Выйти");

while (choice != 4)

{

try

{

Console.Write("Действие: "); choice = Convert.ToInt32(Console.ReadLine());

switch (choice)

{

case 1:

{

Console.Write("Количество: ");

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

studyGroup.Add(mCount);

break;

}

case 2:

{

Console.Write("Количество юношей: "); mCount = Convert.ToInt32(Console.ReadLine());

Console.Write("Количество девушек: "); fCount = Convert.ToInt32(Console.ReadLine());

studyGroup.Deduct(mCount, fCount);

break;

}

case 3:

{

studyGroup.Information();

break;

}

case 4:

{

Console.WriteLine("Пока!");

break;

}

default: Console.WriteLine("Некорректные данные!"); break;

}

}

catch

{

Console.WriteLine("Некорректные данные!");

}

}

Console.ReadLine();

}

public static void ShowMsg(object sender, StudyGroupEventArgs e)

{

Console.WriteLine(e.Message);

}

public static void SortFirst(float[] array)

{

Console.WriteLine("Сортировка");

float mid, sum = 0;

foreach (float item in array)

{

sum += item;

}

mid = sum / array.Length;

Array.Sort(array);

int j = array.Length - 1;

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

{

if (array[i] <= mid)

{

Console.Write($"{array[i]} ");

}

else

{

Console.Write($"{array[j]} ");

j--;

}

}

}

public static void ArrayProcessingFirst(float[] array)

{

Console.WriteLine("Обработка");

char[] ch = { '£', '$', '¥', '€', '¢' };

Random random = new Random();

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

{

if (i == array.Length - 1)

{

array[i] \*= array[0];

break;

}

array[i] \*= array[i + 1];

}

int index = 0;

foreach (float item in array)

{

index = random.Next(0, ch.Length - 1);

Console.Write($"{ch[index]}{item} ");

}

}

public static void NOKSecond(int a, int b)

{

int temp = a > b ? a : b;

int nod = 1;

for (int i = temp; i > 1; i--)

{

if ((a % i == 0) && (b % i == 0))

{

nod = i;

break;

}

}

Console.WriteLine($"НОК: {(a \* b) / nod}");

}

public static void CubeCheckingSecond(int a, int b)

{

int temp = a > b ? a : b;

for (int i = 1; i < temp; i++)

{

if (i \* i \* i == a)

{

Console.WriteLine($"a - куб числа {i}");

}

if (i \* i \* i == b)

{

Console.WriteLine($"b - куб числа {i}");

}

}

}

public static void TrigonomCalcThird(MessageThird mCos, MessageThird mTg, double a, double b)

{

for (double i = a; i <= b; i += 0.4)

{

Console.WriteLine($"cos(2 \* {i,6:F2}) = {mCos(2 \* i),8:F4} \t tg(-5 \* {i,6:F2}) = {mTg(-5 \* i),8:F4}");

}

}

}

}

## Скриншоты:

