Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

Факультет Информационные системы и технологии

**Задача на For** – вариант 14

Отчёт по лабораторной работе № 1

по дисциплине Технология программирования

Выполнил студент ИСТб-20-3 М. Д. Ярков 15.09.2021

номер группы подпись И. О. Фамилия Дата

Принял З. А. Бахвалова . .2021

должность подпись И. О. Фамилия Дата

Иркутск 2021

### Постановка задачи

**Задача №14.** Дана последовательность натуральных чисел. Определить, есть ли в последовательности хотя бы одна n-ка одинаковых “соседних” чисел (n и элементы последовательности вводятся с клавиатуры). В случае положительного ответа определить порядковые номера чисел первой из таких пар.

### Словесное описание алгоритма

1. Создаём числовую переменную n для вводимого количества чисел в последовательности, добавляем в неё полученное от пользователя значение. Число получаем с помощью метода Custom.ReadInt("количество натуральных чисел", 0):
   1. Методу были переданы значения переменным: string name = "количество натуральных чисел", int min = 0, int max = maxInt
   2. Создаём переменную для вывода ошибки при необходимости isInputGot, даём ей значение false – оно обозначает, что пользователь ещё ничего не вводил
   3. Создаём переменную для хранения введённого числа output, даём ей значение unused (константа, введённая для удобства в классе Custom) – оно обозначает, что пользователь ещё ничего не вводил
   4. Запускаем цикл do..while:
      1. Если в isInputGot уже true, выводим сообщение об ошибке
      2. Проверяем значения переменных min и max, которые были переданы методу на шаге 1.1. В зависимости от них выводим соответствующее сообщение, предлагающее пользователю ввести число
      3. Получаем введённую строку, помещаем её в новую переменную input
      4. Задаём переменной isInputGot значение true
      5. Если введённая строка содержит только число, добавляем его в output
   5. Выходим из цикла, если output изменён и входит в заданный промежуток значений
   6. Выводим output
2. Создаём числовую переменную nums для вводимого количества чисел, проверяемых на соседство, заполняем его через Custom.ReadInt("сколько чисел проверять на соседство"
3. Пишем в консоль значение, получаемое из GetN(n, nums):
   1. Методу были переданы n и nums, названия переменных внутри метода идентично
   2. Инициализируем переменные lastNum для последнего числа и sameNums для их количества, задаём значение Custom.Unused обоим
   3. Запускаем цикл, который сработает n раз для перебора чисел, получаемых от пользователя:
      1. В новую переменную newNum добавляем натуральное число с помощью Custom.ReadInt(i + "-ое число", 1)
      2. Проверяем, является ли новое число отличающимся от предыдущего. Если не является, добавляем к sameNums единицу, иначе устанавливаем lastNum значение newNum и sameNums значение 1
      3. Если sameNums равно nums, то искомая последовательность найдена – выводим соотвествующее сообщение
   4. Если вывода не произошло до этого шага (n-ка не найдена), выводим соотвествующее сообщение
4. Вызываем Custom.Exit() для ожидания закрытия программы пользователем:
   1. Пишем в консоль «Для выхода из программы нажмите любую клавишу…»
   2. Ждём нажатия клавиши

### Таблица спецификаций

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| n | Количество чисел в последовательности | Целое число от нуля |
| nums | Количество последовательных чисел, которые необходимо проверить на схожесть | Натуральное число |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Проверка несоседства | 5  2  1 2 1 2 1 | 2-ка чисел не найдена |
| 2 | Проверка соседства | 6  3  1 2 3 3 3 3 | 3-ка чисел 3 найдена в промежутке от 3 до 5 |

Автотесты отсутствуют, поскольку все методы принимают данные через консоль.

### Код программы

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Lab1Num2Var14 {  public class Program {  public static string GetN(int n, int nums) {  int lastNum = Custom.Unused;  int sameNums = Custom.Unused;  for (int i = 1; i <= n; i++) {  int newNum = Custom.ReadInt(i + "-ое число", 1);  if (lastNum == newNum) // число идентично прошлому  sameNums++;  else { // число не идентично прошлому  lastNum = newNum;  sameNums = 1;  }  if (sameNums == nums) // найдена последовательность из n одинаковых чисел  return (nums + "-ка чисел " + lastNum + " найдена в промежутке от " + (i - nums + 1) + " до " + i);  }  return (nums + "-ка чисел не найдена"); // исключение при ненахождении  }  static void Main(string[] args) {  int n = Custom.ReadInt("количество натуральных чисел в последовательности", 0);  int nums = Custom.ReadInt("сколько чисел проверять на соседство", 1);  Console.WriteLine(GetN(n, nums));  Custom.Exit();  }  }  public class Custom {  public const int Unused = Int32.MinValue; // константа для неиспользованных переменных  public const int MinInt = Int32.MinValue; // константа минимального числа для типа Int  public const int MaxInt = Int32.MaxValue; // константа максмального числа для типа Int  public static int ReadInt(string name, int min = MinInt, int max = MaxInt) {  bool isInputGot = false; // переменная для вывода ошибки при необходимости  int output = Unused; // тут будет храниться введённое число  do {  if (isInputGot) // вывод ошибки при повторном вводе  Console.Write("Ошибка ввода! ");  if (min == MinInt && max == MaxInt) // ввод без ОДЗ  Console.Write("Введите " + name + ": ");  else if (min != MinInt && max != MaxInt) // ввод с ОДЗ  Console.Write("Введите " + name + " (в промежутке от " + min + " до " + max + "): ");  else if (min == MinInt) // ввод числа до заданного  Console.Write("Введите " + name + " (до " + max + "): ");  else if (max == MaxInt) // ввод числа от заданного  Console.Write("Введите " + name + " (от " + min + "): ");  string input = Console.ReadLine(); // ввод некоторой строки  isInputGot = true; // изменение переменной для вывода ошибки при необходимости  // перевод строки input в числовой output, если строка является числом  if (int.TryParse(input, out int trash)) output = int.Parse(input);  }  while (output == Unused || output < min || output > max);  return output;  }  public static void Exit() { // ожидание закрытия консоли пользователем  Console.Write("\nДля выхода из программы нажмите любую клавишу...");  Console.ReadKey(true);  }  }  } |

## Группа B

…