|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Волегов Михаил Сергеевич  **Отчет о лабораторной работе №2**  **«Выполнение программы простой структуры. Вычисление математических выражений»**  *Лабораторная работа*  студента образовательной программы «Программная инженерия»  по направлению подготовки *38.03.05 Бизнес-информатика*   |  |  | | --- | --- | |  | Руководитель  Доцент каф. ИТБ, к.п.н.  Ильин Иван Вадимович | |

Пермь, 2021 год

**ПОСТАНОВКА ЗАДАЧИ**

Решить указанные в варианте задачи, используя основные операторы языка C#. При решении задачи, использовать все типы циклов (for, while, do while)

Вариант 3:

1. Дана последовательность из n целых чисел. Найти сумму элементов с четными номерами из этой последовательности. (3)

2. Дана последовательность целых чисел, за которой следует 0. Найти количество элементов этой последовательности, кратных ее первому элементу. (32)

3. S=1+2-3+4+5-6+7+8-9+…, всего n слагаемых. (37)

**АНАЛИЗ**

*Задача 1*

Исходные данные:

n – целое положительное число

число в последовательности – любое целое число

Результаты:

Сумма элементов с четными номерами в последовательности – любое целое число

*Задача 2*

Исходные данные:

число в последовательности – любое целое число

Результаты:

количество чисел в последовательности, кратных первому числу – целое неотрицательное число.

Математическая модель:

X / первое число = 0

*Задача 3*

Исходные данные:

n – количество слагаемых в числовом ряду.

Результаты:

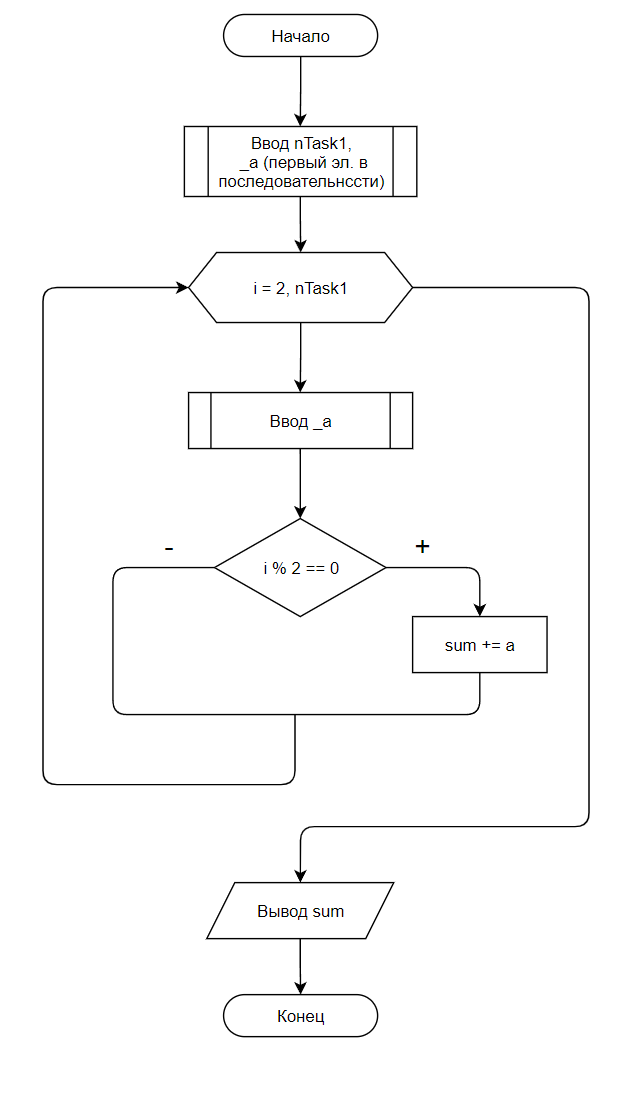
значение числового ряда – любое целое число

Математическая модель:

**ПРОЕКТИРОВАНИЕ**

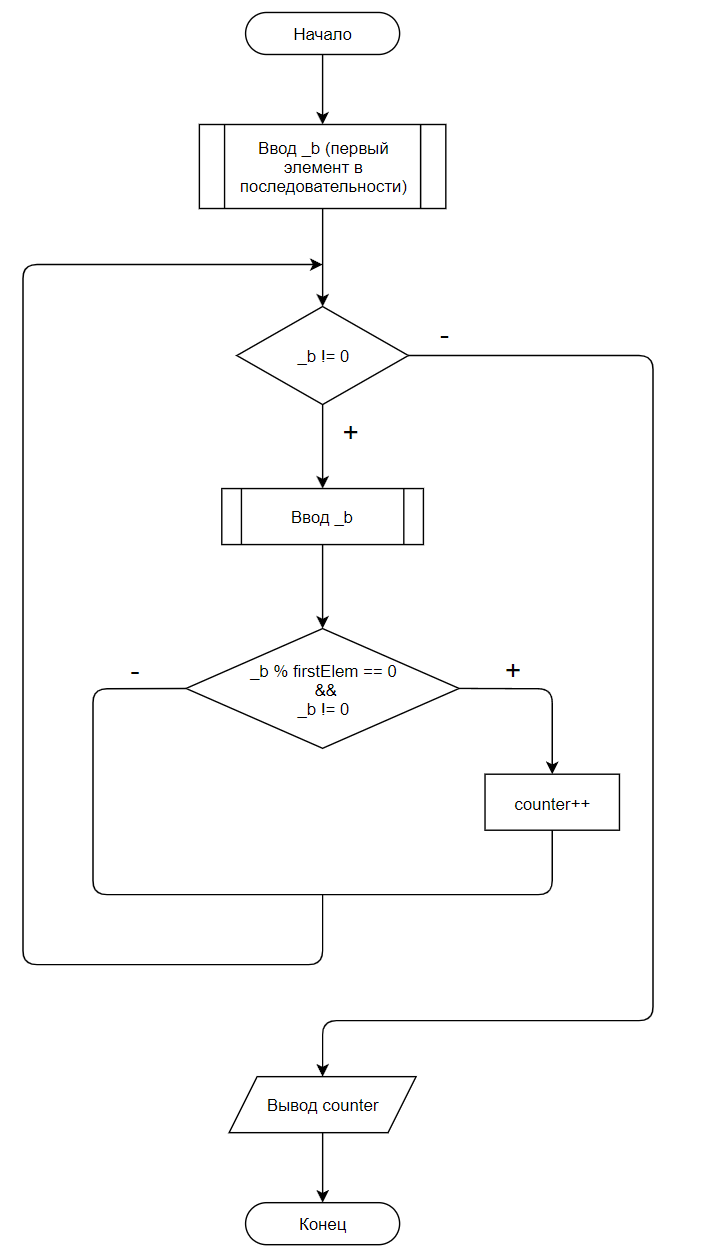
*Задача 1*

Блок-схема



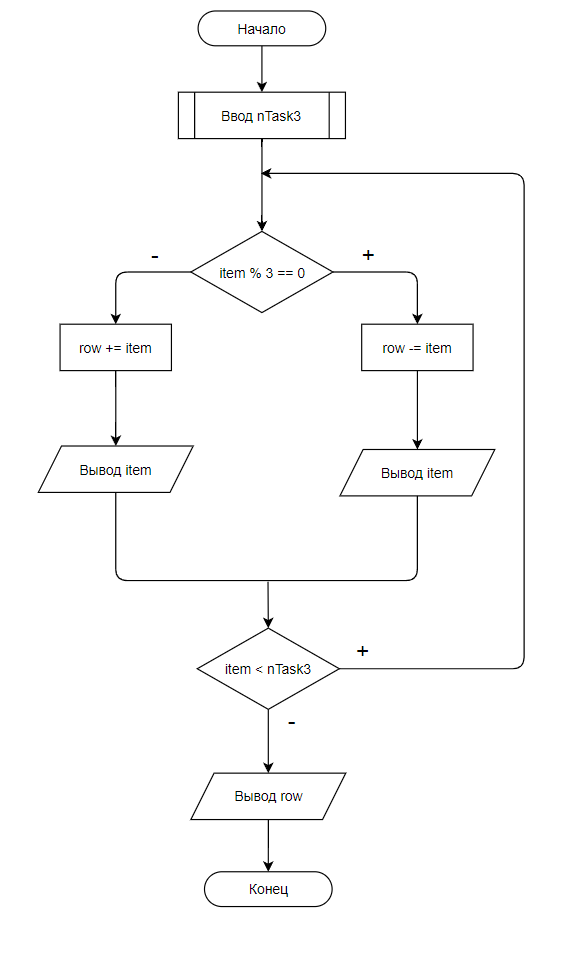
*Задача 2*

Блок-схема

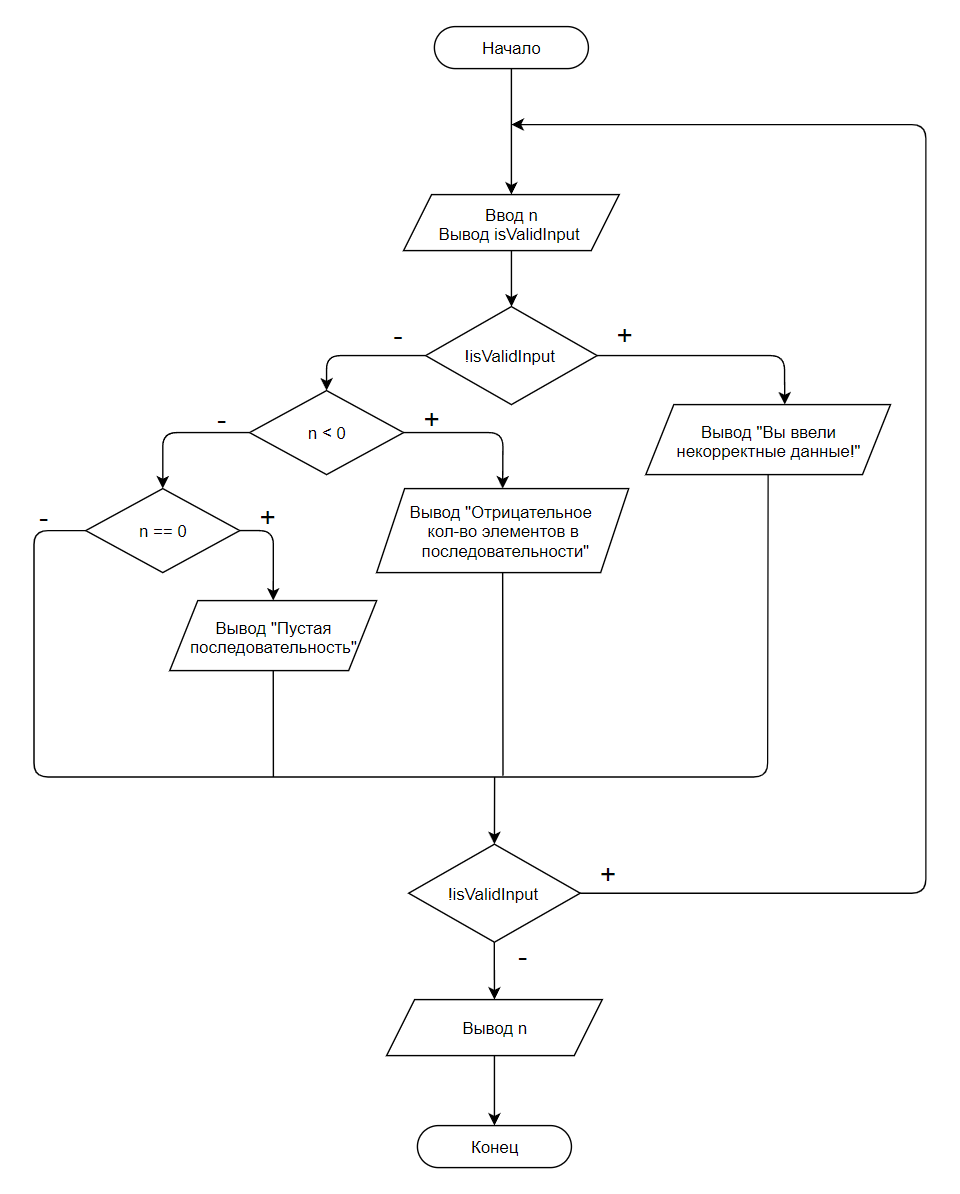
**

*Задача 3*

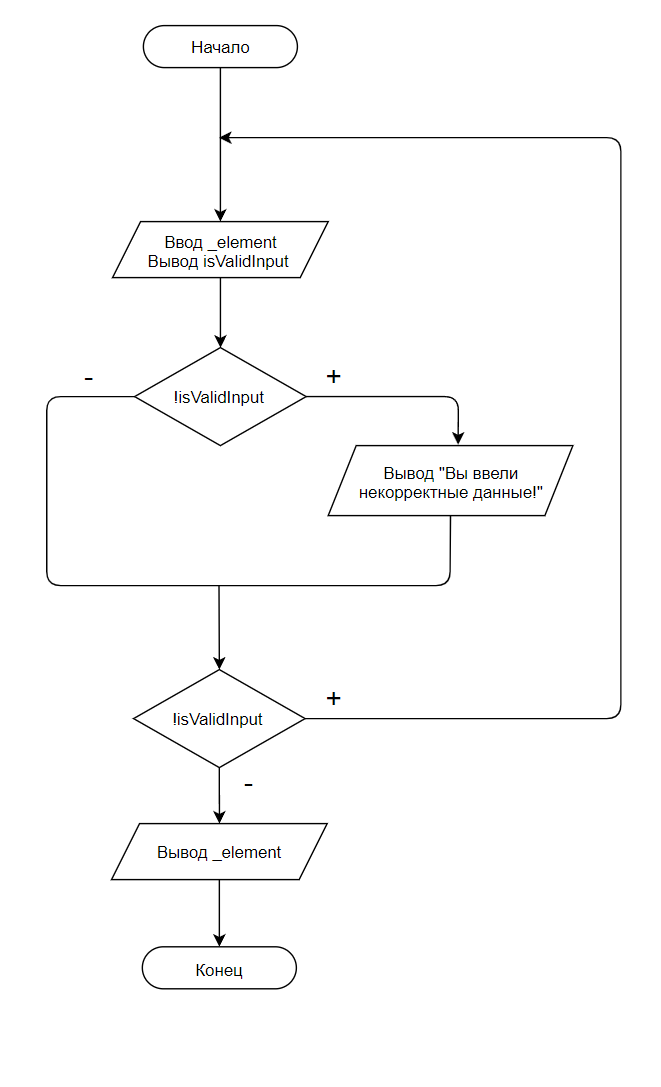
Блок-схема

****

Блок-схема подпрограммы CheckInputSequenceLenght



Блок-схема подпрограммы CheckInputElementInSequence



**ПРОГРАММНЫЙ КОД**

*Задача 1 , 2, 3*

using System;

namespace ConsoleApp1

{

static class Program

{

static void Main(string[] args)

{

Console.WriteLine("Лабораторная работа 2, вариант 3");

Console.WriteLine("\nЗадача 1 (3)");

int sum = 0;

int nTask1 = CheckInputSequenceLenght("n=");

int \_a = CheckInputElementInSequence("a1=");

for (int i = 2; i <= nTask1; i++)

{

\_a = CheckInputElementInSequence($"a{i}=");

if (i % 2 == 0) sum += \_a;

}

Console.WriteLine($"Сумма элементов с четными номерами={sum}");

Console.WriteLine("\nЗадача 2 (32)");

int counter = 0;

int j = 1;

int \_b = CheckInputElementInSequence("1 элемент в последовательности=");

int firstElem = \_b;

while (\_b != 0)

{

j++;

\_b = CheckInputElementInSequence($"{j} элемент в последовательности=");

if (\_b % firstElem == 0 && \_b != 0)

{

counter++;

}

}

Console.WriteLine($"Количество элементов, кратных первому элементу={counter}");

Console.WriteLine("\nЗадача 3 (37)");

int item = 0;

int row = 0;

int nTask3 = CheckInputSequenceLenght("n=");

Console.Write("S=");

do

{

item++;

if (item % 3 == 0)

{

row -= item;

Console.Write("-" + item);

}

else

{

row += item;

Console.Write("+" + item);

}

} while (item < nTask3);

Console.WriteLine($"={row}");

}

static int CheckInputSequenceLenght(string message)

{

bool isValidInput;

int n;

do

{

Console.Write(message);

isValidInput = int.TryParse(Console.ReadLine(), out n);

if (!isValidInput)

{

Console.WriteLine("Вы ввели некорректные данные!");

}

else if (n < 0)

{

Console.WriteLine("Количество элементов в последовательности не может быть отрицательным!");

isValidInput = false;

}

else if (n == 0)

{

Console.WriteLine("Пустая последовательность!");

isValidInput = false;

}

} while (!isValidInput);

return n;

}

static int CheckInputElementInSequence(string message)

{

bool isValidInput;

double \_element;

do

{

Console.Write(message);

isValidInput = int.TryParse(Console.ReadLine(), out \_element);

if (!isValidInput)

{

Console.WriteLine("Вы ввели некорректные данные!");

}

} while (!isValidInput);

return \_element;

}

}

}

**ТЕСТИРОВАНИЕ**

*Задача 1*

Таблица тестов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** | **Примечание** |
| 1 | n=0 | “Пустая последовательность!” | “Пустая последовательность!” | Ввод пустой последовательности |
| 2 | n = -1 | “Количество элементов в последовательности не может быть отрицательным!” | “Количество элементов в последовательности не может быть отрицательным!” | Отрицательное количество элементов в последовательности |
| 3 | n = 2,2 | “Вы ввели некорректные данные” | “Вы ввели некорректные данные” | Нецелое количество элементов |
| 4 | n = 1; ‘a’ | “Вы ввели некорректные данные” | “Вы ввели некорректные данные” | Элемент - символ |
| 5 | n = “asdf” | “Вы ввели некорректные данные” | “Вы ввели некорректные данные” | n - строка |
| 6 | n = ‘a’ | “Вы ввели некорректные данные” | “Вы ввели некорректные данные” | n - символ |
| 7 | n = 1; “sdfsdf” | “Вы ввели некорректные данные” | “Вы ввели некорректные данные” | Элемент - строка |
| 8 | n = 1; 2,2 | “Вы ввели некорректные данные” | “Вы ввели некорректные данные” | Элемент – вещественное число |
| 9 | n = 5; ”1; 2; 3; 4; 5” | “Сумма элементов с четными номерами=6” | “Сумма элементов с четными номерами=6” | Положительная сумма элементов |
| 10 | n = 5; “1; -2; 3; -4; 5” | “Сумма элементов с четными номерами=-6” | “Сумма элементов с четными номерами=-6” | Отрицательная сумма элементов |
| 11 | n = 5; “1; 0; 2; 0; 3” | “Сумма элементов с четными номерами=0” | “Сумма элементов с четными номерами=0” | Сумма элементов = 0 |
| 12 | n = 1; 1 | “Сумма элементов с четными номерами=0” | “Сумма элементов с четными номерами=0” | В последовательности нет элементов с четным номером |
| 13 | n = 2; “1, 2” | “Сумма элементов с четными номерами=2” | “Сумма элементов с четными номерами=2” | Цикл отработает 1 раз |

Таблица критериев ЧЯ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Критерии** | **Т1** | **Т2** | **Т3** | **Т4** | **Т5** | **Т6** | **Т7** | **Т8** | **Т9** | **Т10** | **Т11** | **Т12** | **Т13** |
| **Классы входных данных** | | | | | | | | | | | | |  |
| n – неотрицательное целое число |  |  |  | + |  |  | + | + | + | + | + | + |  |
| n – вещественное число |  |  | + |  |  |  |  |  |  |  |  |  |  |
| n – отрицательное число |  | + |  |  |  |  |  |  |  |  |  |  |  |
| n - нуль | + |  |  |  |  |  |  |  |  |  |  |  |  |
| n - строка |  |  |  |  | + |  |  |  |  |  |  |  |  |
| n - символ |  |  |  |  |  | + |  |  |  |  |  |  |  |
| Эл. Последовательности – целое число |  |  |  |  |  |  |  |  | + | + | + | + | + |
| Эл. Последовательности – вещественное число |  |  |  |  |  |  |  | + |  |  |  |  |  |
| Эл. Последовательности - строка |  |  |  |  |  |  | + |  |  |  |  |  |  |
| Эл. Последовательности - символ |  |  |  | + |  |  |  |  |  |  |  |  |  |
| **Классы выходных данных** | | | | | | | | | | | | |  |
| “Пустая последовательность!” | + |  |  |  |  |  |  |  |  |  |  |  |  |
| “Количество элементов в последовательности не может быть отрицательным!” |  | + |  |  |  |  |  |  |  |  |  |  |  |
| “Вы ввели некорректные данные” |  |  | + | + | + | + | + | + |  |  |  |  |  |
| “Сумма элементов с четными номерами” |  |  |  |  |  |  |  |  | + | + | + | + | + |

Таблица критериев МГТ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 | T12 | Т13 |
| Ввод n (nTask1) | do {} while (!isValidInput) | 1 |  |  |  | + |  |  | + | + | + | + | + | + | + |
| >1 | + | + | + |  | + | + |  |  |  |  |  |  |  |
| If (!isValidInput) | + |  |  | + |  | + | + |  |  |  |  |  |  |  |
| - | + | + |  | + |  |  | + | + | + | + | + | + | + |
| If (n < 0) | + |  | + |  |  |  |  |  |  |  |  |  |  |  |
| - | + |  | + | + | + | + | + | + | + | + | + | + | + |
| If (n == 0) | + | + |  |  |  |  |  |  |  |  |  |  |  |  |
| - |  | + | + | + | + | + | + | + | + | + | + | + | + |
| Ввод элемента в последовательности | do {} while (!isValidInput) | 1 |  |  |  |  |  |  |  |  | + | + | + | + | + |
| >1 |  |  |  | + |  |  | + | + |  |  |  |  |  |
| If (!isValidInput) | + |  |  |  | + |  |  | + | + |  |  |  |  |  |
| - |  |  |  |  |  |  |  |  | + | + | + | + | + |
| Основной цикл | For (int i = 2; I < nTask1; i++) | 0 |  |  |  |  |  |  |  |  |  |  |  | + |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  | + |
| >1 |  |  |  |  |  |  |  |  | + | + | + |  |  |

*Задача 2*

Таблица тестов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** | **Примечание** |
| 1 | 0 | “Количество элементов, кратных первому элементу=0” | “Количество элементов, кратных первому элементу=0” | Первый элемент 0 |
| 2 | 5; 6; 15; 7; 25; 0 | “Количество элементов, кратных первому элементу=2” | “Количество элементов, кратных первому элементу=2” | 2 элемента, кратных первому элементу |
| 3 | 5; 1; 2; 3; 4; 0 | “Количество элементов, кратных первому элементу=0” | “Количество элементов, кратных первому элементу=0” | Нет элементов, кратных первому элементу |
| 4 | 5; 15; 20; 25; 0 | “Количество элементов, кратных первому элементу=3” | “Количество элементов, кратных первому элементу=3” | Все элементы в последовательности кратны первому |
| 5 | ‘a’ | “Вы ввели некорректные данные!” | “Вы ввели некорректные данные!” | Элемент - символ |
| 6 | “asdf” | “Вы ввели некорректные данные!” | “Вы ввели некорректные данные!” | Элемент - строка |
| 7 | 6,6 | “Вы ввели некорректные данные!” | “Вы ввели некорректные данные!” | Элемент – вещественное число |
| 8 | 1; 2; 0 | “Количество элементов, кратных первому элементу=1” | “Количество элементов, кратных первому элементу=1” | Цикл выполнится 1 раз |

Таблица критериев ЧЯ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Критерии** | **Т1** | **Т2** | **Т3** | **Т4** | **Т5** | **Т6** | **Т7** |
| **Классы входных данных** | | | | | | | |
| Первый эл. последовательности - нуль | + |  |  |  |  |  |  |
| Эл. Последовательности – целые числа |  | + | + | + |  |  |  |
| Эл. Последовательности - символ |  |  |  |  | + |  |  |
| Эл. Последовательности - строка |  |  |  |  |  | + |  |
| Эл. Последовательности – вещественное число |  |  |  |  |  |  | + |
| **Классы выходных данных** | | | | | | | |
| “Количество элементов, кратных первому элементу” | + | + | + | + |  |  |  |
| “Вы ввели некорректные данные!” |  |  |  |  | + | + | + |

Таблица критериев МГТ

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Т1** | **Т2** | **Т3** | **Т4** | **Т5** | **Т6** | **Т7** | **Т8** |
| Ввод элемента последовательности (\_b) | do {} while (!isValidInput) | 1 | + | + | + | + |  |  |  |  |
| >1 |  |  |  |  | + | + | + |  |
| If (!isValidInput) | + |  |  |  |  | + | + | + |  |
| - | + | + | + | + |  |  |  |  |
| Основной цикл | While (\_b != 0) | 0 | + |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  | + |
| >1 |  | + | + | + |  |  |  |  |
| Проверка условия | If (\_b % firstElem == 0 && \_b != 0) | + |  | + |  | + |  |  |  | + |
| - |  |  | + |  |  |  |  |  |

*Задача 3*

Таблица тестов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** | **Примечание** |
| 1 | n = 9 | “S=+1+2-3+4+5-6+7+8-9=9” | “S=+1+2-3+4+5-6+7+8-9=9” | 9 слагаемых |
| 2 | n = 1 | “S=+1=1” | “S=+1=1” | 1 слагаемое |
| 3 | n = 0 | “Пустая последовательность!” | “Пустая последовательность!” | 0 слагаемых |
| 4 | n = ‘a’ | “Вы ввели некорректные данные!” | “Вы ввели некорректные данные!” | n - символ |
| 5 | n = “asdf” | “Вы ввели некорректные данные!” | “Вы ввели некорректные данные!” | n - строка |
| 6 | n = 2,2 | “Вы ввели некорректные данные!” | “Вы ввели некорректные данные!” | n – вещественное число |
| 7 | n = -1 | “Количество элементов в последовательности не может быть отрицательным!” | “Количество элементов в последовательности не может быть отрицательным!” | n – отрицательное число |

Таблица критериев ЧЯ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Критерии** | **Т1** | **Т2** | **Т3** | **Т4** | **Т5** | **Т6** | **Т7** |
| **Классы входных данных** | | | | | | | |
| n – целое число | + | + |  |  |  |  | + |
| n - нуль |  |  | + |  |  |  |  |
| n - символ |  |  |  | + |  |  |  |
| n - строка |  |  |  |  | + |  |  |
| n – вещественное число |  |  |  |  |  | + |  |

Таблица критериев МГТ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Т1** | **Т2** | **Т3** | **Т4** | **Т5** | **Т6** | **Т7** |
| Ввод n (nTask3) | do {} while (!isValidInput) | 1 | + | + |  |  |  |  |  |
| >1 |  |  | + | + | + | + | + |
| If (!isValidInput) | + |  |  |  | + | + | + |  |
| - | + | + | + |  |  |  | + |
| If (n < 0) | + |  |  |  |  |  |  | + |
| - | + | + | + | + | + | + |  |
| If (n == 0) | + |  |  | + |  |  |  |  |
| - | + | + |  | + | + | + | + |
| Основной цикл | do {} while (item < nTask3) | 1 |  | + |  |  |  |  |  |
| >1 | + |  |  |  |  |  |  | **Примечание** |
| Проверка условия | If (item % 3 == 0) | + | + |  |  |  |  |  |  | T1 – выполнится ветка TRUE и ветка FALSE |
| - | + | + |  |  |  |  |  |  |