Министерство образования Московской области

**ГБПОУ МО «Серпуховский колледж»**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ УП.01**

Специальность 09.02.07 Информационные системы и программирование

Руководитель практики:

Черникова Л.В.

«» 20\_\_г.

*(подпись)*

Исполнитель:

Студент Шойбеков Р.Ж.,

Курс 2, группа 1222

«» 20\_\_г.

*(подпись)*

Серпухов, 2024 г.

СОДЕРЖАНИЕ

Оглавление

[Специальность 09.02.07 Информационные системы и программирование 0](#_Toc168472423)

[Исполнитель: 0](#_Toc168472424)

[СОДЕРЖАНИЕ 1](#_Toc168472425)

[ВВЕДЕНИЕ 6](#_Toc168472426)

[1. Задание первое 7](#_Toc168472427)

[1.1. Цель работы 7](#_Toc168472428)

[1.2. Задание к работе 7](#_Toc168472429)

[1.3. Постановка задачи 7](#_Toc168472430)

[1.3.1. Математическая модель задачи 8](#_Toc168472431)

[1.3.2. Входные данные 8](#_Toc168472432)

[1.3.3. Промежуточные данные 8](#_Toc168472433)

[1.3.4. Выходные данные 8](#_Toc168472434)

[1.3.5. Логическая модель задачи 9](#_Toc168472435)

[1.4. Текст программы 10](#_Toc168472436)

[1.5. Тестовые данные 11](#_Toc168472437)

[1.6. Результаты работы программы 12](#_Toc168472438)

[2. Задание ВТОРОЕ 13](#_Toc168472439)

[2.1. Цель работы 13](#_Toc168472440)

[2.2. Задание к работе 13](#_Toc168472441)

[2.3. Постановка задачи 13](#_Toc168472442)

[2.3.1. Математическая модель задачи 14](#_Toc168472443)

[2.3.2. Входные данные 14](#_Toc168472444)

[2.3.3. Промежуточные данные 14](#_Toc168472445)

[2.3.4. Выходные данные 15](#_Toc168472446)

[2.3.5. Логическая модель задачи 15](#_Toc168472447)

[2.4. Текст программы 20](#_Toc168472448)

[2.5. Тестовые данные 22](#_Toc168472449)

[2.6. Результаты работы программы 23](#_Toc168472450)

[3. Задание ТРЕТЬЕ 24](#_Toc168472451)

[3.1. Цель работы 24](#_Toc168472452)

[3.2. Задание к работе 24](#_Toc168472453)

[3.3. Постановка задачи 24](#_Toc168472454)

[3.3.1. Математическая модель задачи 25](#_Toc168472455)

[3.3.2. Входные данные 25](#_Toc168472456)

[3.3.3. Промежуточные данные 26](#_Toc168472457)

[3.3.4. Выходные данные 26](#_Toc168472458)

[3.3.5. Логическая модель задачи 27](#_Toc168472459)

[3.4. Текст программы 32](#_Toc168472460)

[3.5. Тестовые данные 34](#_Toc168472461)

[3.6. Результаты работы программы 34](#_Toc168472462)

[4. Задание ЧЕТВЁРТОЕ 35](#_Toc168472463)

[4.1. Цель работы 35](#_Toc168472464)

[4.2. Задание к работе 35](#_Toc168472465)

[4.3. Постановка задачи 35](#_Toc168472466)

[4.3.1. Математическая модель задачи 35](#_Toc168472467)

[4.3.2. Входные данные 35](#_Toc168472468)

[4.3.3. Промежуточные данные 35](#_Toc168472469)

[4.3.4. Выходные данные 35](#_Toc168472470)

[4.3.5. Логическая модель задачи 35](#_Toc168472471)

[4.4. Текст программы 36](#_Toc168472472)

[4.5. Тестовые данные 36](#_Toc168472473)

[4.6. Результаты работы программы 36](#_Toc168472474)

[5. Задание ПЯТОЕ 37](#_Toc168472475)

[5.1. Цель работы 37](#_Toc168472476)

[5.2. Задание к работе 37](#_Toc168472477)

[5.3. Постановка задачи 37](#_Toc168472478)

[5.3.1. Математическая модель задачи 37](#_Toc168472479)

[5.3.2. Входные данные 37](#_Toc168472480)

[5.3.3. Промежуточные данные 37](#_Toc168472481)

[5.3.4. Выходные данные 37](#_Toc168472482)

[5.3.5. Логическая модель задачи 37](#_Toc168472483)

[5.4. Текст программы 38](#_Toc168472484)

[5.5. Тестовые данные 38](#_Toc168472485)

[5.6. Результаты работы программы 38](#_Toc168472486)

[6. Задание ШЕСТОЕ 39](#_Toc168472487)

[6.1. Цель работы 39](#_Toc168472488)

[6.2. Задание к работе 39](#_Toc168472489)

[6.3. Постановка задачи 39](#_Toc168472490)

[6.3.1. Математическая модель задачи 39](#_Toc168472491)

[6.3.2. Входные данные 39](#_Toc168472492)

[6.3.3. Промежуточные данные 39](#_Toc168472493)

[6.3.4. Выходные данные 39](#_Toc168472494)

[6.3.5. Логическая модель задачи 39](#_Toc168472495)

[6.4. Текст программы 40](#_Toc168472496)

[6.5. Тестовые данные 40](#_Toc168472497)

[6.6. Результаты работы программы 40](#_Toc168472498)

[Заключение 41](#_Toc168472499)

[список используемых источников 42](#_Toc168472500)

[приложение 1 – какое-то приложение 43](#_Toc168472501)

ВВЕДЕНИЕ

Целью данной учебной практики является разработка приложения для работы с разными типами данных, решение поставленных задач через подпрограммы, повторение изученного материала.

Задачи:

– повторение пройденного материала;

– грамотное использование всех знаний для всевозможного облегчения создания программы;

– повторение синтаксиса программы, использование различных классов, методов и типов данных;

1. Задание первое
   1. Цель работы

Цель работы: получить практические навыки разработки линейных программ с использованием Windows Form в среде Visual Studio.

* 1. Задание к работе

1. Формализовать задачу
2. Разработать алгоритм решения задачи
3. Разработать пользовательскую форму, предназначенную для расчёта рентабельности
4. Разработать код программы
5. Протестировать программу
   1. Постановка задачи

Выполнить расчёт рентабельности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сумма  Выручки(В) | Затраты на  Производство(ЗП) | Затраты на  Реализацию(ЗР) | Себестоимость (С) | Рентабельность (Р) |
| 78 618,00 р | 50 000,00 р | 20 000,00 р | (ЗП+ЗР) | ([В-С]/С\*100) |

* + 1. Математическая модель задачи

Себестоимость вычисляется по формуле:

С = ЗП + ЗР, (1.1)

где ЗП – затраты на производство,

ЗР – затраты на реализацию.

Рентабельность вычисляется по формуле:

Р = ([В-С]/С\*100) , (1.2)

где В – сумма выручки,

С – себестоимость.

* + 1. Входные данные

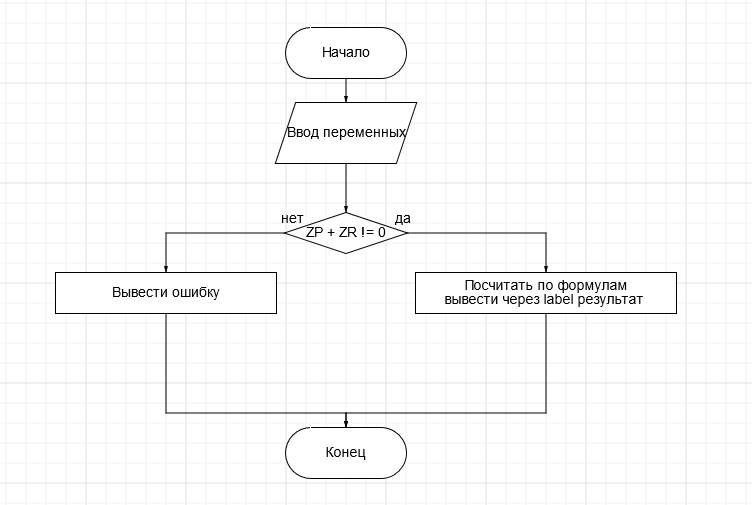
|  |  |  |  |
| --- | --- | --- | --- |
| Наименование переменной | Идентификатор | Тип данных | Ограничение |
| Сумма  Выручки | summV | double | 2.22507 x 10-308 и 1.79769 x 10308 |
| Затраты на  Производство | ZP | double | 2.22507 x 10-308 и 1.79769 x 10308 |
| Затраты на  Реализацию | ZR | double | 2.22507 x 10-308 и 1.79769 x 10308 |

* + 1. Промежуточные данные
    2. Выходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование переменной | Идентификатор | Тип данных | Ограничение |
| Рентабельность | profitability | double | 2.22507 x 10-308 и 1.79769 x 10308 |

* + 1. Логическая модель задачи

Рисунок 1.1.1



* 1. Текст программы

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace Практика\_Шойбеков\_24.\_06\_обновленный\_от\_0.\_4.\_06\_  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void button1\_Click(object sender, EventArgs e)  {  // Получение значений из текстовых полей  double summV = Convert.ToDouble(textBox1.Text); // Суммарные доходы  double ZP = Convert.ToDouble(textBox2.Text); // Зарплата  double ZR = Convert.ToDouble(textBox3.Text); // Зарплата руководителя  // Проверка, не равно ли C нулю  if (ZP + ZR != 0)  {  double C = ZP + ZR; // Общая сумма зарплаты  double Rent = ((summV - C) / C \* 100); // Процент рентабельности  // Вывод результатов в соответствующие метки  label6.Text = Convert.ToString(C);  label7.Text = Convert.ToString(Rent);  }  else  {  // Вывод сообщения об ошибке, если C равно нулю  MessageBox.Show("Общая сумма зарплаты не может быть равна нулю!");  }  }  private void textBox1\_TextChanged(object sender, EventArgs e)  {  }  private void textBox2\_TextChanged(object sender, EventArgs e)  {  }  private void textBox3\_TextChanged(object sender, EventArgs e)  {  }  private void label6\_Click(object sender, EventArgs e)  {  }  private void label7\_Click(object sender, EventArgs e)  {  }  private void button2\_Click(object sender, EventArgs e)  {  // Очистка текстовых полей и меток  textBox1.Text = "";  textBox2.Text = "";  textBox3.Text = "";  label6.Text = "";  label7.Text = "";  }  }  } |

* 1. Тестовые данные

|  |  |  |
| --- | --- | --- |
| Номер теста | Вводимые значения | Результат выполнения программы |
| 1  2  3 | double summV = 23  double ZP = 7  double ZR = 5 | 12  91,6 |

* 1. Результаты работы программы

Рисунок 1.1.2 – изображен ввод данных с последующим расчетом

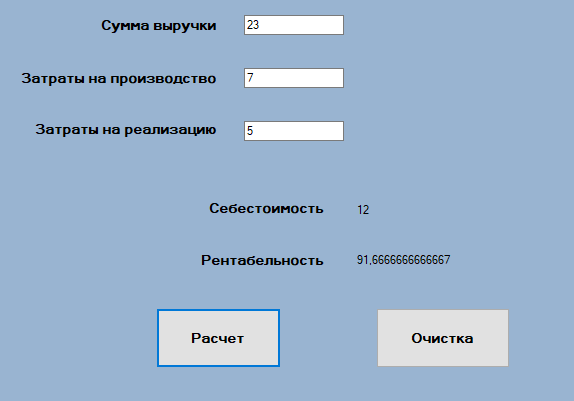
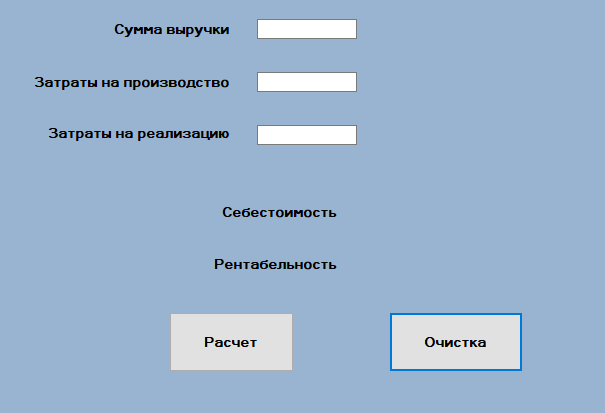


Рисунок 1.1.3 – изображена очистка данных



1.7 Вывод

Вывод: мы получить практические навыки разработки программ с применением файлов и с использованием Windows Form в среде Visual Studio

1. Задание ВТОРОЕ
   1. Цель работы

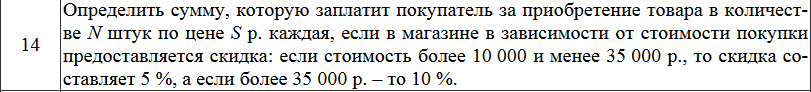
Цель работы: получить практические навыки разработки программ с линейной структурой с использованием Windows Form в среде Visual Studio.

* 1. Задание к работе

1. Формализовать задачу
2. Разработать алгоритм решения задачи
3. Разработать пользовательский интерфейс
4. Разработать код программы
5. Протестировать программу
6. Привести результаты программы
7. Сделать вывод о проделанной работе
   1. Постановка задачи

Выполнение расчёта суммы заказа приведено на рисунке 1.1

Рисунок 1.1.3



* + 1. Математическая модель задачи

1. Рассчитываем общую стоимость покупки без скидки: total = price \* quantity
2. Определяем скидку в зависимости от общей стоимости покупки:
   1. если total находится в диапазоне от 10000 до 35000, то скидка составляет 5% (discount = 0.05)
   2. если total превышает 35000, то скидка составляет 10% (discount = 0.10)
3. Рассчитываем итоговую стоимость покупки с учётом скидки: finalPrice = total - (total \* discount)
   * 1. Входные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Идентификатор | Тип данных | Ограничение |
| Цена | price | double | 2.22507 x 10-308 и 1.79769 x 10308 |
| Количество | quantity | int | Только положительные, целые значения |

* + 1. Промежуточные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Идентификатор | Тип данных | Ограничение |
| Массив(цена и кол-во товара) | array | double | 2.22507 x 10-308 и 1.79769 x 10308 |
| Общ.стоимость без скидки | total | double | 2.22507 x 10-308 и 1.79769 x 10308 |
| Скидка | discount | double | 2.22507 x 10-308 и 1.79769 x 10308 |

* + 1. Выходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Идентификатор | Тип данных | Ограничение |
| Итоговая стоимость | finalPrice | double | 2.22507 x 10-308 и 1.79769 x 10308 |

* + 1. Логическая модель задачи

Структура программы представлена на рисунке 1.1.4

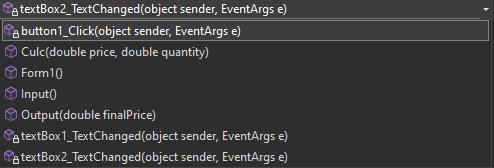


Рис. 1.1.4 – структура программы

Применяемые методы: button1\_Click – метод нажатия кнопки для вывода итоговой стоимости покупки; Culc – метод для расчета итоговой стоимости покупки; Input – метод для ввода значений под цену и количества товара; Output – метод для создания MessageBox с последующим сообщением о итоговой стоимости.

Схема алгоритма метода Culc представлена на рисунке 1.1.5

Рисунок 1.1.5

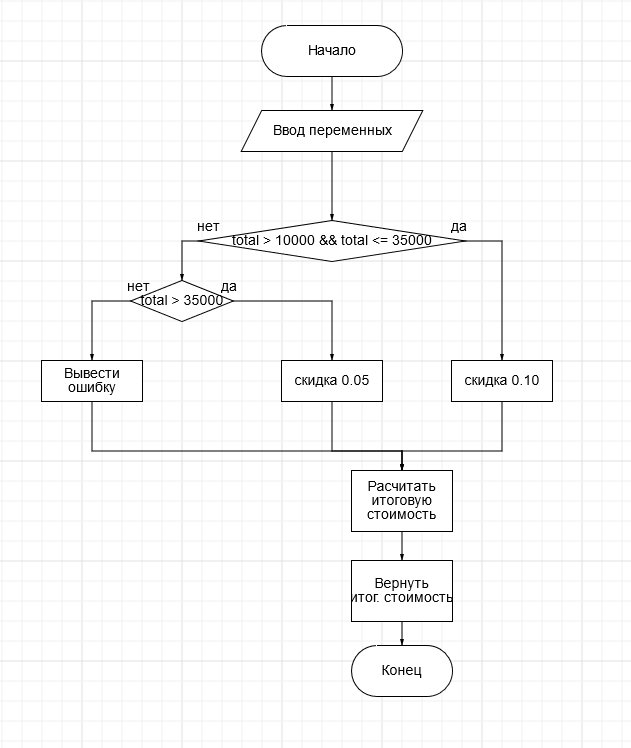


Схема алгоритма метода Input представлена на рисунке 1.1.6

Рисунок 1.1.6

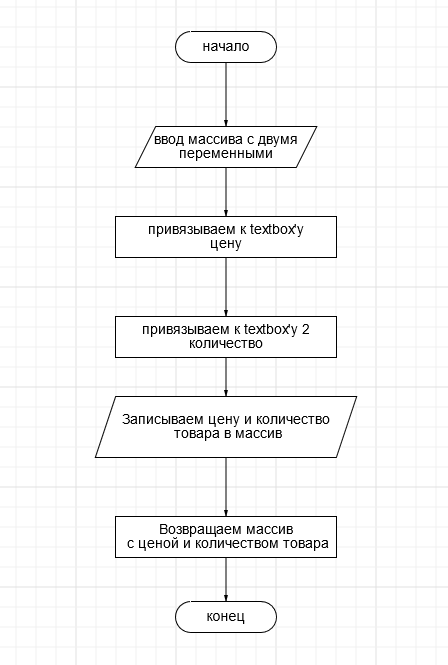


Схема алгоритма метода Output представлена на рисунке 1.1.7

Рисунок 1.1.7



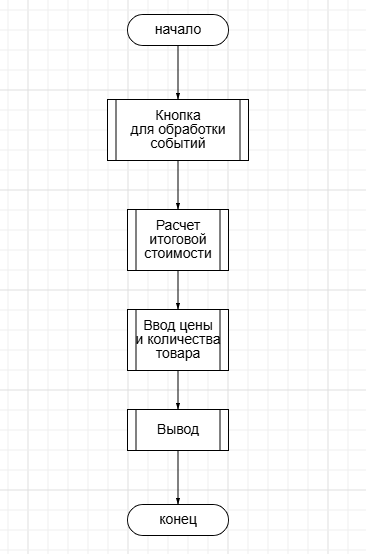
Схема алгоритма метода button1\_Click представлена на рисунке 1.1.8

Рисунок 1.1.8



Линейная структура программы 1.1.9

Рисунок 1.1.9



* 1. Текст программы

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace Практика\_Шойбеков\_2805\_обновленный\_04.\_06\_  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void button1\_Click(object sender, EventArgs e)  {  // Вводим цену и количество товара  double[] array = Input();  // Рассчитываем итоговую стоимость покупки  double finalPrice = Culc(array[0], array[1]);  // Выводим итоговую стоимость покупки  Output(finalPrice);  }  static int[] InputResidents()  {  int[] residents = new int[12];  for (int i = 0; i < 12; i++)  {  Console.Write($"Введите число жителей в доме {i + 1}: ");  residents[i] = Convert.ToInt32(Console.ReadLine());  }  return residents;  }  /// Рассчитывает итоговую стоимость покупки  public double Culc(double price, double quantity)  {  // Рассчитываем общую стоимость покупки без скидки  double total = price \* quantity;  // Инициализируем скидку нулём  double discount = 0;  // Определяем скидку в зависимости от общей стоимости покупки  if (total > 10000 && total <= 35000)  {  discount = 0.05;  }    else if (total > 35000)  {  discount = 0.10;  }    else  {  MessageBox.Show("Ошибка при вводе данных");  }  // Рассчитываем итоговую стоимость покупки с учётом скидки  double finalPrice = total - (total \* discount);  // Возвращаем итоговую стоимость покупки  return finalPrice;  }  ///Массив из двух элементов: цены и количества товара  public double[] Input()  {  double[] array = new double[2];  // Получаем цену товара  double price = int.Parse(textBox1.Text);  // Получаем количество товара  int quantity = int.Parse(textBox2.Text);  // Записываем цену и количество товара в массив  array[0] = price;  array[1] = Convert.ToInt32(quantity);  // Возвращаем массив с ценой и количеством товара  return array;  }  ///Итоговая стоимость покупки  public void Output(double finalPrice)  {  // Выводим итоговую стоимость покупки в MessageBox  MessageBox.Show($"Общая сумма покупки {finalPrice} руб.");  }  private void textBox1\_TextChanged(object sender, EventArgs e)  {  }  private void textBox2\_TextChanged(object sender, EventArgs e)  {  }  }  } |

* 1. Тестовые данные

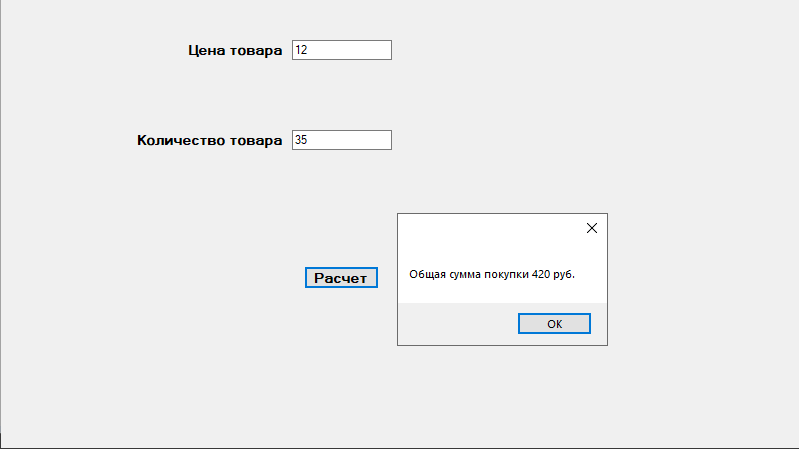
Тестовые данные представлены в таблице 1.5.

Таблица 1.5 – тестовые данные

|  |  |  |
| --- | --- | --- |
| Номер теста | Вводимые значения | Результат выполнения программы |
| 1  2 | Price = 12  Quantity = 35 | 420 |

* 1. Результаты работы программы

Рисунок 1.2



2.7 Вывод

Вывод: мы получить практические навыки разработки программ с применением файлов и с использованием Windows Form в среде Visual Studio

1. Задание ТРЕТЬЕ
   1. Цель работы

Цель работы: получить практические навыки разработки программ с линейной структурой с использованием Windows Form в среде Visual Studio

* 1. Задание к работе

1) Формализовать задачу

2) Разработать алгоритм решения задачи

3) Разработать пользовательскую форму, предназначенную для расчёта рентабельности

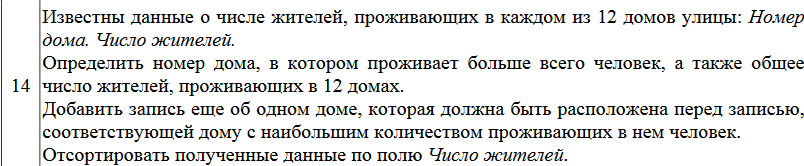
4) Разработать код программы

5) Протестировать программу

* 1. Постановка задачи

Выполнение расчёта суммы заказа приведено на рисунке 1.2.1

Рисунок 1.2.1



* + 1. Математическая модель задачи

Определите функцию InputResidents, которая не принимает аргументов и возвращает массив целых чисел, представляющих количество жителей в каждом из 12 домов. Эта функция предлагает пользователю ввести количество жителей в каждом доме и сохраняет их в массиве.

Определите функцию CalculateTotalResidents, которая принимает массив целых чисел в качестве аргумента и возвращает единственное целое число, представляющее общее количество жителей во всех домах. Эта функция инициализирует переменную totalResidents значением 0 и выполняет итерацию по массиву, добавляя количество жителей в каждом доме к totalResidents.

Определите функцию FindMaxResidentsHouse, которая принимает массив целых чисел в качестве аргумента и возвращает целое число, представляющее номер дома с максимальным числом жителей. Эта функция инициализирует переменные maxResidents и maxResidentsHouse первым элементом массива и его индексом соответственно. Затем она выполняет итерацию по остальному массиву, обновляя maxResidents и maxResidentsHouse

* + 1. Входные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Идентификатор | Тип данных | Ограничение |
| Количество жителей | residents | Int[] | Только положительные, целые значения |

* + 1. Промежуточные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Идентификатор | Тип данных | Ограничение |
| Массив целых чисел, представляющих количество жителей в каждом доме | residents | int[] | Только положительные, целые значения |
| Общее количество жителей во всех домах | totalResident | int | Только положительные, целые значения |

* + 1. Выходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Идентификатор | Тип данных | Ограничение |
| Общее количество жителей во всех домах. | totalResidents | Int | Только положительные, целые значения |
| Номер дома с максимальным количеством жителей. | maxResidentsHouse | int | Только положительные, целые значения |

* + 1. Логическая модель задачи

Структура программы представлена на рисунке 1.2.2

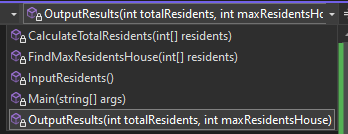


Рис. 1.2.2 – структура программы

Применяемые методы: CalculateTotalResidents – Метод для расчета общего числа жителей; FindMaxResidentsHouse – Метод для поиска номера дома с максимальным числом жителей; InputResidents – Метод для ввода числа жителей в каждом доме; OutputResults –Метод для вывода результатов

Схема алгоритма метода CalculateTotalResidents() представлена на рисунке 1.2.3

Рисунок 1.2.3

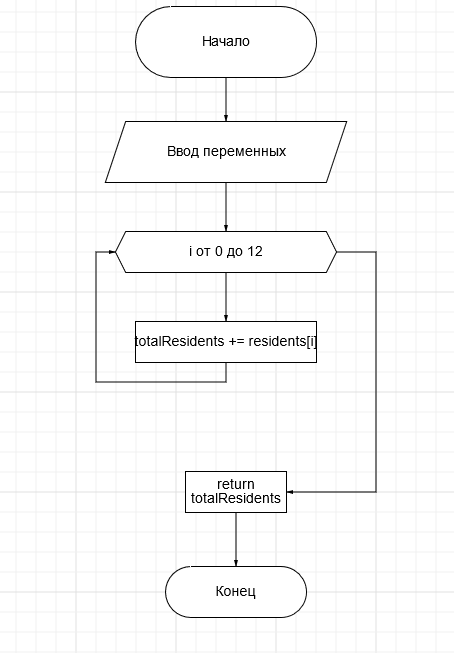


Схема алгоритма метода FindMaxResidentsHouse() представлена на рисунке 1.2.4

Рисунок 1.2.4

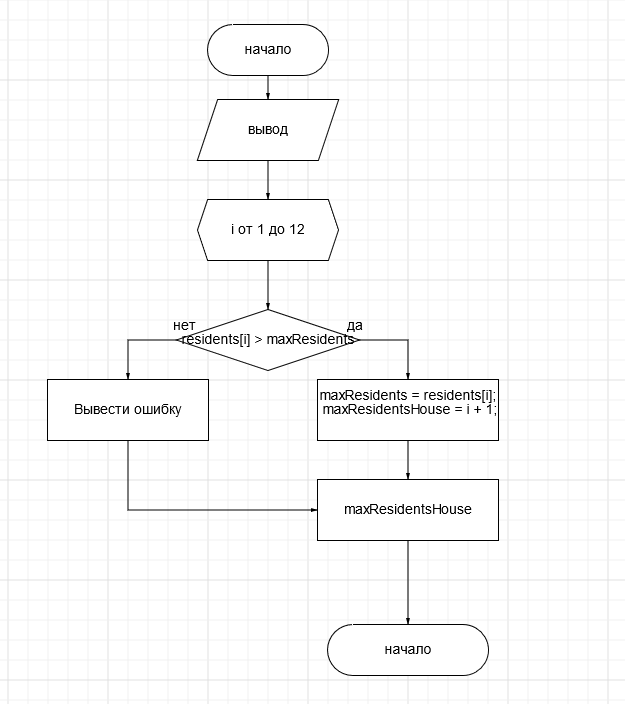


Схема алгоритма метода InputResidents() представлена на рисунке 1.2.5

Рисунок 1.2.5

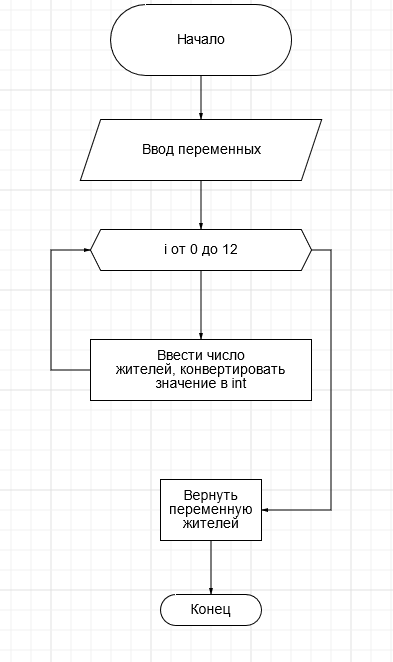
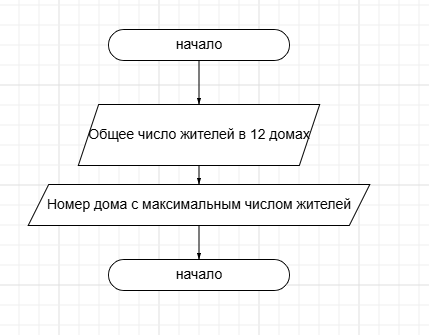


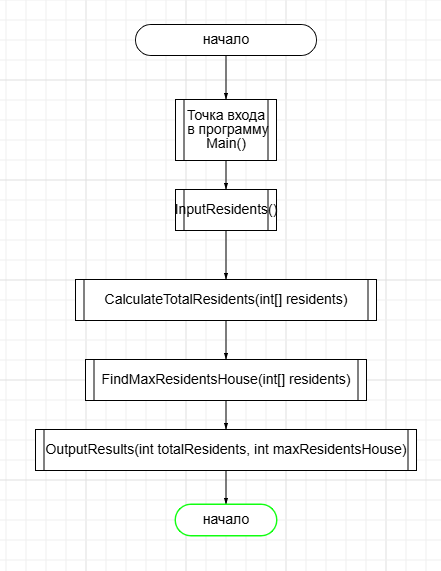
Схема алгоритма метода OutputResults() представлена на рисунке 1.2.6

Рисунок 1.2.6



Структура всей программы на рисунке 1.2.7

Рисунок 1.2.7



* 1. Текст программы

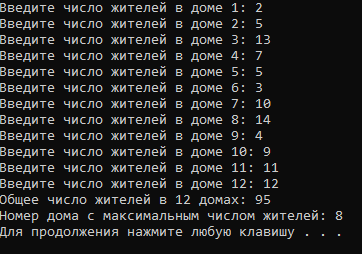
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Практика\_Шойбеков\_28.\_05\_  {  internal class Program  {  static void Main(string[] args)  {  // Вводим число жителей в каждом доме  int[] residents = InputResidents();  // Рассчитываем общее число жителей  int totalResidents = CalculateTotalResidents(residents);  // Находим номер дома с максимальным числом жителей  int maxResidentsHouse = FindMaxResidentsHouse(residents);  // Выводим результаты  OutputResults(totalResidents, maxResidentsHouse);  }  ///Массив с числом жителей в каждом доме.</returns>  static int[] InputResidents()  {  int[] residents = new int[12];  for (int i = 0; i < 12; i++)  {  Console.Write($"Введите число жителей в доме {i + 1}: ");  residents[i] = Convert.ToInt32(Console.ReadLine());  }  return residents;  ///Общее число жителей  static int CalculateTotalResidents(int[] residents)  {  int totalResidents = 0;  for (int i = 0; i < 12; i++)  {  totalResidents += residents[i];  }  return totalResidents;  }  /// Находит номер дома с максимальным числом жителей.  static int FindMaxResidentsHouse(int[] residents)  {  int maxResidents = residents[0];  int maxResidentsHouse = 1;  for (int i = 1; i < 12; i++)  {  if (residents[i] > maxResidents)  {  maxResidents = residents[i];  maxResidentsHouse = i + 1;  }  }  return maxResidentsHouse;  }  /// Выводит результаты.  static void OutputResults(int totalResidents, int maxResidentsHouse)  {  Console.WriteLine($"Общее число жителей в 12 домах: {totalResidents}");  Console.WriteLine($"Номер дома с максимальным числом жителей: {maxResidentsHouse}");  }  }  } |

* 1. Тестовые данные

|  |  |  |
| --- | --- | --- |
| Номер теста | Вводимые значения | Результат выполнения программы |
| 1 | Int residents ={2,5,13,7,5,3,10,14,4,9,11,12} | 95  8 |

* 1. Результаты работы программы

Рисунок 1.2.8 – на рисунке изображена консоль с вводом и выводом итоговых данных



3.7 Вывод

Вывод: мы получить практические навыки разработки программ с применением файлов и с использованием Windows Form в среде Visual Studio

1. Задание ЧЕТВЁРТОЕ
   1. Цель работы
   2. Задание к работе
   3. Постановка задачи
      1. Математическая модель задачи
      2. Входные данные
      3. Промежуточные данные
      4. Выходные данные
      5. Логическая модель задачи
   4. Текст программы
   5. Тестовые данные
   6. Результаты работы программы
2. Задание ПЯТОЕ
   1. Цель работы
   2. Задание к работе
   3. Постановка задачи
      1. Математическая модель задачи
      2. Входные данные
      3. Промежуточные данные
      4. Выходные данные
      5. Логическая модель задачи
   4. Текст программы
   5. Тестовые данные
   6. Результаты работы программы
3. Задание ШЕСТОЕ
   1. Цель работы
   2. Задание к работе
   3. Постановка задачи
      1. Математическая модель задачи
      2. Входные данные
      3. Промежуточные данные
      4. Выходные данные
      5. Логическая модель задачи
   4. Текст программы
   5. Тестовые данные
   6. Результаты работы программы

Заключение

В результате выполнения практики все поставленные мною цели были достигнуты.

список используемых источников

приложение 1 – какое-то приложение