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

Отчёт по лабораторной работе № 2

по дисциплине «Технология программирования»

Выполнил студент ИСТб-20-3 А.В.Имихелов 05.10.2021

номер группы подпись И. О. Фамилия Дата

Принял З. А. Бахвалова . .2021

должность подпись И. О. Фамилия Дата

Иркутск 2021

# Лабораторное задание №2. GUI – интерфейс.

## Задача на If

### Постановка задачи

**Задача №13. Поезд прибывает на станцию в a часов b минут и отправляется в c часов d минут. Пассажир пришел на платформу в n часов m минут. Будет ли поезд стоять на платформе? Числа a, b, c, d, n, m — целые, 0 < a 23, 0 < b 59, 0 < c 23, 0 < d 59, 0 < n 23, 0 < m 59**

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Назначение | Параметры | Тип |
| Методы класса “Form1” | | | |
| Form1\_FormClosed | Форма интерфейса | object sender, EventArgs e | void |
| button1\_Click | Кнопка «Рассчитать» | object sender, EventArgs e | void |
| button2\_Click | Кнопка «Очистить» | object sender, EventArgs e | void |
| Form1\_Load | Форма интерфейса | object sender, EventArgs e | void |
| a\_KeyDown | Перенос поля ввода | object sender, EventArgs e | void |
| b\_KeyDown | Перенос поля ввода | object sender, EventArgs e | void |
| c\_KeyDown | Перенос поля ввода | object sender, EventArgs e | void |
| d\_KeyDown | Перенос поля ввода | object sender, EventArgs e | void |
| n\_KeyDown | Перенос поля ввода | object sender, EventArgs e | void |
| m\_KeyDown | Перенос поля ввода | object sender, EventArgs e | void |
| Методы класса “Logic” | | | |
| compare | Логика программы | A,b,c,d,n,m | void |
| isNumber | Проверка строки на то, является ли она числом | a | boolean |
| MoreThanHour | Метод, необходимые для проверки времени остановки поезда | A,b,c,d | boolean |
| LowThanOne | Метод, необходимые для проверки времени остановки поезда | A,b,c,d | boolean |

### Таблица спецификаций

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| a | Время прибытия поезда в часах | Действительное положительное число |
| b | Время прибытия поезда в минутах | Действительное положительное число |
| c | Время отбытия поезда в часах | Действительное положительное число |
| d | Время отбытия поезда в минутах | Действительное положительное число |
| n | Время прибытия пассажира в часах | Действительное положительное число |
| m | Время прибытия пассажира в минутах | Действительное положительное число |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Некорректный ввод | \*пустые поля ввода\* | «Введите значение» |
| 2 | Некорректный ввод | A=63, b =15, c=23, d=22, n=22, m=15 | «Внимание! Неверно задано число.» |
| 3 | Корректный ввод | A=15, b =20, c=15, d=40, n=15, m=35 | «Поезд не на станции!» |

### Код 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 Lab2ex1  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void button2\_Click(object sender, EventArgs e)  {  a.Clear();  b.Clear();  c.Clear();  d.Clear();  n.Clear();  m.Clear();  resultBox.Clear();  }  private void Form1\_Load(object sender, EventArgs e)  {  a.Text = Properties.Settings.Default.a\_save;  b.Text = Properties.Settings.Default.b\_save;  c.Text = Properties.Settings.Default.c\_save;  d.Text = Properties.Settings.Default.d\_save;  n.Text = Properties.Settings.Default.n\_save;  m.Text = Properties.Settings.Default.m\_save;  }  private void button1\_Click(object sender, EventArgs e) //кнопка "Рассчитать"  {  int detector = 0;  int a\_int=0, b\_int=0, c\_int=0, d\_int=0, n\_int=0, m\_int = 0;  if ((a.Text == "") || (b.Text == "") || (c.Text == "") || (d.Text == "") || (n.Text == "") || (m.Text == ""))  MessageBox.Show("Введите значение!");  else  {  if (Logic.isNumber(Convert.ToString(a.Text)) == false)  {  MessageBox.Show("Введите числовое значение!");  detector = 1;  }  else  {  a\_int = Convert.ToInt32(a.Text);  if ((a\_int > 23) || (a\_int < 0))  {  MessageBox.Show("Внимание! Неверно задано число!");  detector = 1;  }  }  if (Logic.isNumber(Convert.ToString(b.Text)) == false)  {  MessageBox.Show("Введите числовое значение!");  detector = 1;  }  else  {  b\_int = Convert.ToInt32(b.Text);  if ((b\_int > 59) || (b\_int < 0))  {  MessageBox.Show("Внимание! Неверно задано число!");  detector = 1;  }  }  if (Logic.isNumber(Convert.ToString(c.Text)) == false)  {  MessageBox.Show("Введите числовое значение!");  detector = 1;  }  else  {  c\_int = Convert.ToInt32(c.Text);  if ((c\_int > 23) || (c\_int < 0))  {  MessageBox.Show("Внимание! Неверно задано число!");  detector = 1;  }  }  if (Logic.isNumber(Convert.ToString(d.Text)) == false)  {  MessageBox.Show("Введите числовое значение!");  detector = 1;  }  else  {  d\_int = Convert.ToInt32(d.Text);  if ((d\_int > 59) || (d\_int < 0))  {  MessageBox.Show("Внимание! Неверно задано число!");  detector = 1;  }  }  if (Logic.isNumber(Convert.ToString(n.Text)) == false)  {  MessageBox.Show("Введите числовое значение!");  detector = 1;  }  else  {  n\_int = Convert.ToInt32(n.Text);  if ((n\_int > 23) || (n\_int < 0))  {  MessageBox.Show("Внимание! Неверно задано число!");  detector = 1;  }  }  if (Logic.isNumber(Convert.ToString(m.Text)) == false)  {  MessageBox.Show("Введите числовое значение!");  detector = 1;  }  else  {  m\_int = Convert.ToInt32(m.Text);  if ((m\_int > 59) || (m\_int < 0))  {  MessageBox.Show("Внимание! Неверно задано число!");  detector = 1;  }  }  if (Logic.MoreThanHour(a\_int, b\_int, c\_int, d\_int) == true)  {  MessageBox.Show("Поезд не может стоять более одного часа! Повторите ввод.");  detector = 1;  }  if (Logic.LowThanOne(a\_int, b\_int, c\_int, d\_int) == true)  {  MessageBox.Show("Поезд не может стоять менее минуты! Повторите ввод.");  detector = 1;  }  if (detector == 0)  {  var OutMessage = Logic.Compare(a\_int, b\_int, c\_int, d\_int, n\_int, m\_int);  resultBox.Text = OutMessage;  }  }  }  private void Form1\_FormClosed(object sender, FormClosedEventArgs e) //сохранение  {  Properties.Settings.Default.a\_save = a.Text;  Properties.Settings.Default.b\_save = b.Text;  Properties.Settings.Default.c\_save = c.Text;  Properties.Settings.Default.d\_save = d.Text;  Properties.Settings.Default.n\_save = n.Text;  Properties.Settings.Default.m\_save = m.Text;  Properties.Settings.Default.Save();  }  private void a\_KeyDown(object sender, KeyEventArgs e) //перенос поля ввода по нажатию enter  {  if (e.KeyCode == Keys.Enter)  b.Focus();  }  private void b\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  c.Focus();  }  private void c\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  d.Focus();  }  private void d\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  n.Focus();  }  private void n\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  m.Focus();  }  private void m\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  button1.Focus();  }  }  public class Logic  {  public static Boolean isNumber(string a) //метод, необходимый для корректности ввода пользователя  {  int detector = 0;  for (int i = 0; i < a.Length; i++)  {  if (((int)a[i] < 48) || ((int)a[i] > 57)) //цикл, проверяющий каждый элемент, вводимый пользователем  detector++;  }  if (detector == 0)  return true;  return false;  }  public static Boolean MoreThanHour(int a, int b, int c, int d) //методы, необходимые для проверки времени остановки поезда  {  int minutes\_ab = minutes(a, b);  int minutes\_cd = minutes(c, d);  if (minutes\_cd-minutes\_ab>=60)  return true;  return false;  }  public static int minutes(int h, int m)  {  return h \* 60 + m;  }  public static Boolean LowThanOne(int a, int b, int c, int d)  {  if (((c - a == 0) && (d - b == 0)))  return true;  return false;  }  public static string Compare(int a, int b, int c, int d, int n, int m) //метод, отвечающий за основную логику программы  {  int minutes\_ab = minutes(a, b);  int minutes\_cd = minutes(c, d);  int minutes\_nm = minutes(n, m);  string OutMessage = "";  if (minutes\_ab <= minutes\_nm && minutes\_nm <= minutes\_cd)  OutMessage = "Поезд на станции!";  else  OutMessage = "Поезд не на станции!";  return OutMessage;  }  }  } |

# Результат

Результаты выполнения тестов с 1 по 4 приведены на рисунках с 1 по 4 соответственно.

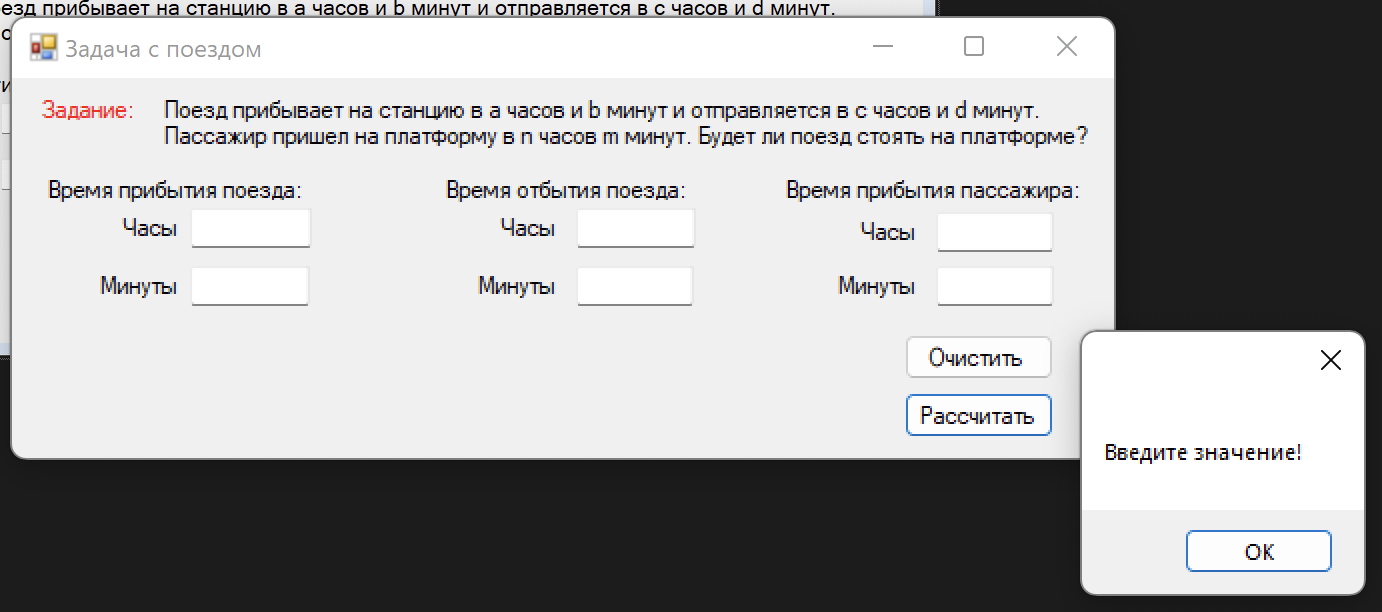


Рисунок 1 — Результат выполнения теста 1.

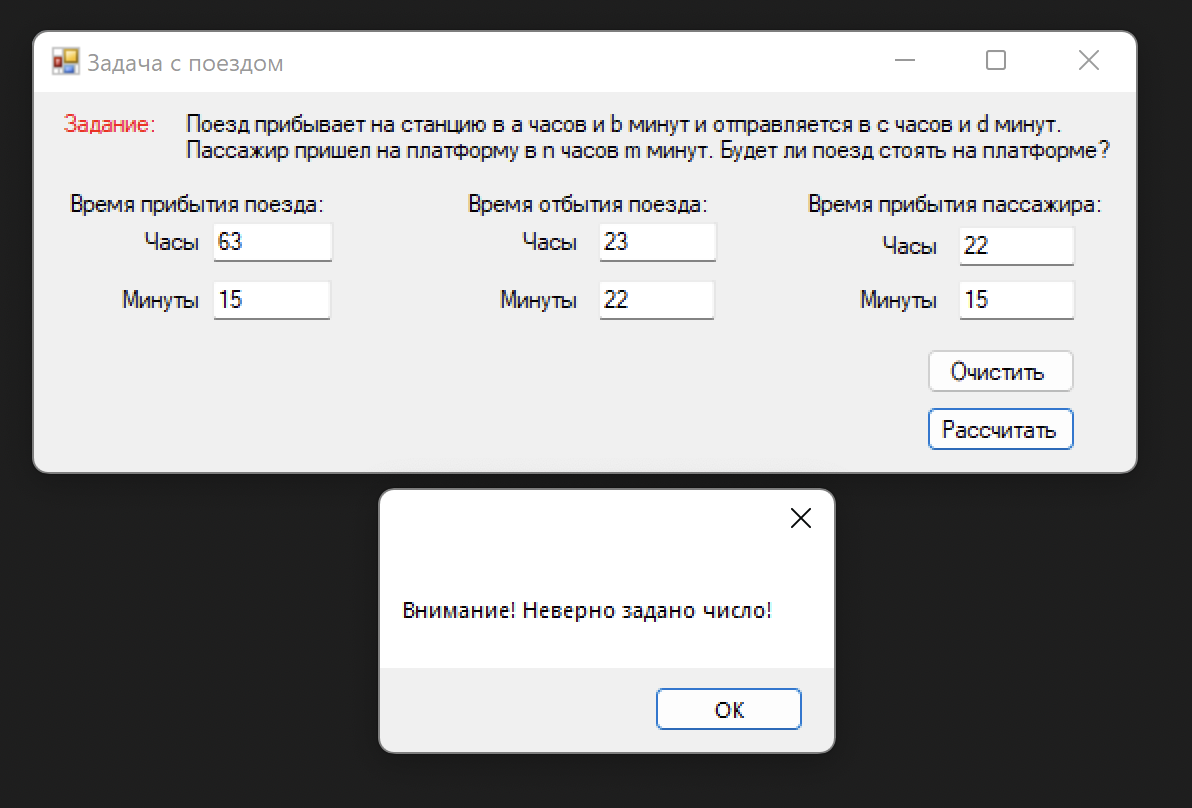


Рисунок 2 — Результат выполнения теста 2.

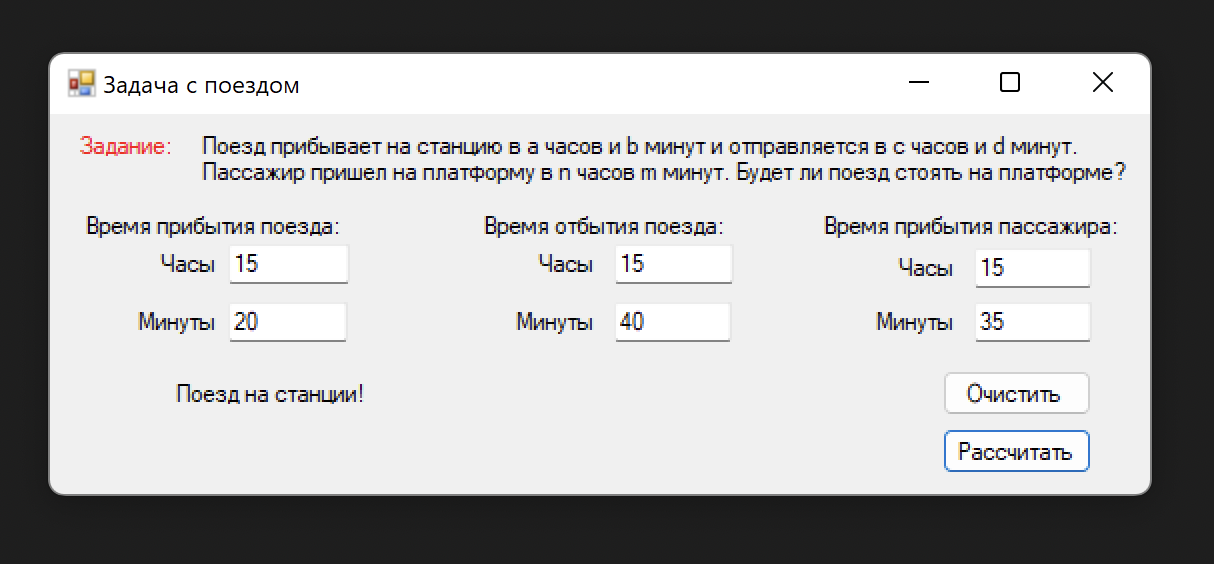


Рисунок 3 — Результат выполнения теста 3.

### Код автотестов

|  |
| --- |
| using NUnit.Framework;  using Train;  namespace Test  {  public class Tests  {  [SetUp]  public void Setup()  {  }  [Test]  public void TrainTestStay()  {  int a = 13;  int b = 55;  int c = 14;  int d = 00;  int n = 13;  int m = 56;  var message = Logic.Compare(a, b, c, d, n, m);  Assert.AreEqual("Поезд на станции!", message);  }  [Test]  public void TrainTestAway()  {  int a = 13;  int b = 55;  int c = 14;  int d = 00;  int n = 14;  int m = 05;  var message = Logic.Compare(a, b, c, d, n, m);  Assert.AreEqual("Поезд не на станции!", message);  }  }  } |