**Федеральное агентство связи**

**Государственное бюджетное образовательное учреждение высшего**

**образование**

**Ордена Трудового Красного Знамени**

**«Московский технический университет связи и информатики»**

Кафедра информатики

**Отчет по лабораторной работе №4**

«Разработка программы с разветвлениями

“АРМ оператора обменного пункта”»

по дисциплине «Информатика»

Подготовил студент

группы БВТ1903: Щитов В.М.

Проверил: Волков А.И.

Москва 2019

Оглавление

[1 Постановка задачи 3](#_Toc27971238)

[1.1 Общее задание на разработку программы 3](#_Toc27971239)

[1.2 Требования к элементам пользовательского интерфейса 4](#_Toc27971240)

[1.3 Требования к организации ввода-вывода данных в программе 9](#_Toc27971241)

[1.4 Требования к реакции программы на нестандартные, в том числе критические, действия пользователя 9](#_Toc27971242)

[1.5 Прочие требования к программе 9](#_Toc27971243)

[2 Схемы алгоритмов 10](#_Toc27971244)

[3 Код программы 16](#_Toc27971245)

[4 Результаты тестирования программы 20](#_Toc27971246)

[Вывод 22](#_Toc27971247)

# **1 Постановка задачи**

## **Общее задание на разработку программы**

Разработать на Visual C++ приложение «АРМ оператора обменного пункта». Условие задачи предполагает следующие пункты его реализации:

1) Создать форму, имеющую вид, представленный на рисунке 1, приблизительный результат работы которой также представлен на рисунке 1.

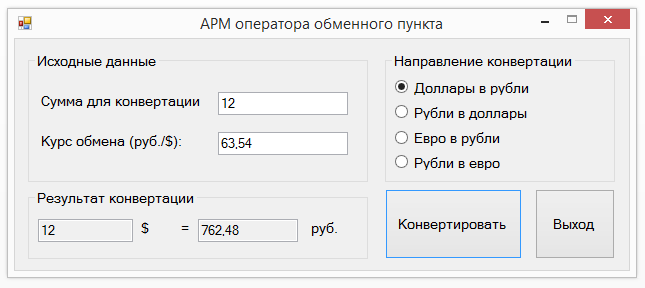


Рисунок 1 – Окно приложения «АРМ оператора обменного пункта».

2) Реализовать введение числовых значений в поля типа TextBox, . ограничить ввод букв и специальных символов, при этом должен корректно вводиться знак десятичного разделителя для обеспечения ввода вещественных чисел.

3) При нажатии на кнопку «Конвертировать» должно создаваться событие обрабатываемое входные данные в соответствии с алгоритмом конвертации валют.

4) При изменении направления конвертации должны «очищаться» поля результатов работы, а единицы измерения должны меняться.

5) Если не введены исходные данные (сумма для конвертации или курс обмена) или введены недопустимые значения, то должны выдаваться соответствующие сообщения.

6) Введённый курс обмена в каждом направлении должен запоминаться и появляться при очередном выборе данного направления обмена.

7) При получении вещественных значений конвертированной валюты результат должен округляться до сотых.

8) Нажатие кнопки «Выход» должно осуществлять корректное завершение работы приложения.

## **1.2 Требования к элементам пользовательского интерфейса**

Для данной программы необходимо на форме расположить 2 элемента типа Button, выполняющих операцию конвертации и закрытие программы, 8 меток типа Label, поясняющих содержимое соответствующих им элементов типа TextBox, 4 элемента типа TextBox, 2 из которых предназначены для ввода преобразуемой суммы и текущего курса валют, а 2 для вывода преобразуемого и преобразованного значений, 4 элемента типа RadioButton для переключения между типами конвертируемых значений, 3 элемента типа GroupBox для группирования и систематизации других элементов формы. Описание свойств формы представлено в таблице 1.

Таблица 1 – Свойства формы и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| Text | Заголовок формы | АРМ оператора обменного пункта |
| FormBorderStyle | Стиль обрамления  формы | FixedDialog (неизменяемый размер окна) |
| MaximizeBox | Доступность кнопки «Развернутость» в заголовке окна | False (кнопка «Развернуть» недоступна) |
| Size.Width | Ширина формы | 630 |
| Size.Height | Высота формы | 270 |

Свойства элементов управления типа Button, к которым относятся btnAction и btnClose, отвечающие за конвертацию валют и закрытие приложения, представлены в таблице 2.

Таблица 2 – Свойства кнопок и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя кнопки | btnAction, btnClose |
| Text | Заголовок кнопки | «Конвертировать», «Выход» |
| Location X | Расстояние от левого края | 370, 520 |
| Location Y | Расстояние от верхнего края | 150 |
| Size.Width | Ширина кнопки | 137, 80 |
| Size.Height | Высота кнопки | 70 |

Свойства элементов управления Label, к которым относятся label1, lblExchangeRate, lblFrom, label4, lblTo, предназначенные для пояснения назначения элементов типа TextBox и типов конвертируемых валют, представлены в таблице 3.

Таблица 3 – Свойства меток и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя метки | label1, lblExchangeRate, lblFrom, label4, lblTo |
| Text | Текст метки | «Сумма для конвертации», «Курс обмена (руб./$)», «руб.», «=», «$», «Результат конвертации» |
| TextAlign | Способ выравнивания текста | TopLeft |
| Location X | Расстояние от левого края | 10, 110, 150, 280 |
| Location Y | Расстояние от верхнего края | 40, 80, 30 |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – обычный  Размер – 10 pt. |

Свойства элементов управления типа TextBox, к которым относятся txt1, txt2, txtResult1 и txtResult2, предназначенные для ввода преобразуемой суммы и заданного курса и вывода преобразуемого и преобразованного значений, представлены в таблице 4.

Таблица 4 – Свойства элементов типа TextBox и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Namе) | Название | txt1, txt2, txtResult1, txtResult2 |
| Sizе | Размер: длина и высота | (130; 23), (100; 23) |
| Location | Координаты (x; y) | (190; 40, 80),  (10, 170; 30) |
| TextAlign | Задает выравнивание текста в полях редактирования | Left |
| RеadOnly | Задает, может ли вводиться значение с клавиатуры | False, True |

Свойства элементов управления типа RadioButton, к которым относятся radioBtn1, radioBtn2, radioBtn3, radioBtn4, отвечающие за выбор направления конвертации валют, представлены в таблице 5.

Таблица 5 – Свойства элементов типа RadioButton и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя кнопки | radioBtn1, radioBtn2, radioBtn3, radioBtn4 |
| Text | Заголовок кнопки | «Доллары в рубли», «Рубли в доллары», «Евро в рубли», «Рубли в евро» |
| Location | Координаты (x; y) | (13; 13), (13; 150),  (370; 13) |
| Size.Width | Ширина группы | 340, 340, |
| Size.Height | Высота группы | 130, 170, |

Свойства элементов управления типа GroupBox, к которым относятся groupBox1, groupBox2, groupBox3, отвечающие за группирование другие элементов формы в отдельным группы, представлены в таблице 6.

Таблица 6 – Свойства элементов типа GroupBox и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя группы | groupBox1, groupBox2, groupBox3 |
| Text | Заголовок группы | «Исходные данные», «Результат конвертации», «Направление конвертации» |
| Location X | Расстояние от левого края | 10 |
| Location Y | Расстояние от верхнего края | 25, 50, 75, 100 |
| Size.Width | Ширина группы | 340, 340, 230 |
| Size.Height | Высота группы | 130, 70, 130 |

В таблице 7 указаны элементы управления, принадлежащие к той или иной группе.

Таблица 7 – Управляющие элементы, принадлежащие элементам типа GroupBox и их значения

|  |  |
| --- | --- |
| **Группа** | **Элементы группы** |
| groupBox1 | label1, lblExchangeRate, txt1, txt2 |
| groupBox2 | txtResult1, txtResult2, label 4, lblTo, lblFrom |
| groupBox3 | radioBtn1, radioBtn2, radioBtn3, radioBtn4 |

## **1.3 Требования к организации ввода-вывода данных в программе**

Ввод данных должен осуществляться с клавиатуры, вывод должен производиться на экран компьютера.

## **1.4 Требования к реакции программы на нестандартные, в том числе критические, действия пользователя**

Недопустим ввод любых данных кроме числовых и десятичного разделителя.

## **Прочие требования к программе**

Программа должна работать с вещественными числами, корректно реагировать на неправильный ввод требуемых для расчётов значений пользователем. Нажатие на кнопку «Выход» должно корректно завершать работу программы.

# **2 Схемы алгоритмов**

На рисунке 2 изображена блок-схема события txt1\_TextChanged, отвечающего за ввод корректных данных в текстовое поле txt1.



Рисунок 2 – Схема алгоритма функции txt1\_TextChanged

На рисунке 3 изображена блок-схема события txt2\_TextChanged, отвечающего за ввод корректных данных в текстовое поле txt2.



Рисунок 3 – Схема алгоритма функции txt2\_TextChanged

На рисунке 4 изображена блок-схема события btnClose\_Click, отвечающего за корректное закрытие программы.



Рисунок 4 – Схема алгоритма функции btnClose\_Click

На рисунке 5 изображена блок-схема события txt2\_Changed, отвечающего за сохранение значения курса валюты при переключении на другой курс.



Рисунок 5 – Схема алгоритма функции txt2\_Changed

На рисунках 6, 7, 8 и 9 изображены блок-схемы событий radioBtn1\_CheckedChanged, radioBtn2\_CheckedChanged, radioBtn3\_CheckedChanged и radioBtn4\_CheckedChanged соответственно, отвечающие за изменение значения курса валюты и внешнего вида формы при переключении на заданный курс.



Рисунок 6 – Схема алгоритма функции radioBtn1\_CheckedChanged



Рисунок 7 – Схема алгоритма функции radioBtn2\_CheckedChanged



Рисунок 8 – Схема алгоритма функции radioBtn3\_CheckedChanged



Рисунок 9 – Схема алгоритма функции radioBtn4\_CheckedChanged

На рисунке 10 изображена блок-схема события btnAction\_Click, отвечающего за сохранение выполнение конвертированию валюты по нажатию на кнопку «Конвертировать».



Рисунок 10 – Схема алгоритма функции btnAction\_Click

# **3 Код программы**

Исходный код программы представлен в ниже приведённом листинге.

//Обработчик для кнопки «Выход»

private: System::Void btnClose\_Click(System::Object^ sender, System::EventArgs^ e) {

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "Вы уверены, что хотите выйти?", "Выход", MessageBoxButtons::YesNo, MessageBoxIcon::Exclamation, MessageBoxDefaultButton::Button2);

if (result == System::Windows::Forms::DialogResult::Yes) {

this->Close();

}

}

private:

int x1;

double x3;

//переменные для валюты

double RubToDol = 0.016, DolToRub = 63.54, RubToEuro = 0.014, EuroToRub = 70.45;

//Обработчик ввода текста в первое текстовое поле

private: System::Void txt1\_TextChanged(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

if ((!Char::IsDigit(e->KeyChar)) && (e->KeyChar != ',' || txt1->Text->IndexOf(',') != -1) && (e->KeyChar != (char)Keys::Back)) e->Handled = true;

if ((e->KeyChar == (char)Keys::D0 || e->KeyChar == (char)Keys::NumPad0) && (txt1->Text->Length >= 1) && (txt1->Text[0] == '0' && txt1->SelectionStart < 2)) e->Handled = true;

if (e->KeyChar == ',') {

x1 = 0;

if (txt1->Text->IndexOf(',') == -1 && txt1->SelectionStart == x1) {

txt1->Text = txt1->Text->Insert(x1, "0");

txt1->SelectionStart = x1 + 1;

}

}

}

//Обработчик ввода текста во второе текстовое поле

private: System::Void txt2\_TextChanged(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

if ((!Char::IsDigit(e->KeyChar)) && (e->KeyChar != ',' || txt2->Text->IndexOf(',') != -1) && (e->KeyChar != (char)Keys::Back)) e->Handled = true;

if ((e->KeyChar == (char)Keys::D0 || e->KeyChar == (char)Keys::NumPad0) && (txt2->Text->Length >= 1) && (txt2->Text[0] == '0' && txt2->SelectionStart < 2)) e->Handled = true;

if (e->KeyChar == ',') {

x1 = 0;

if (txt2->Text->IndexOf(',') == -1 && txt2->SelectionStart == x1) {

txt2->Text = txt2->Text->Insert(x1, "0");

txt2->SelectionStart = x1 + 1;

}

}

}

//Обработчик сохранения курсов валюи при переключении на другой курс

private: System::Void txt2\_Changed(System::Object^ sender, System::EventArgs^ e) {

if (txt2->Text !=(Convert::ToString(DolToRub)) && radioBtn1->Checked == true) {

if (txt2->Text == "" || txt2->Text == ",") {

DolToRub = 0;

}

else {

DolToRub = Convert::ToDouble(txt2->Text);

}

}

else if (txt2->Text != (Convert::ToString(RubToDol)) && radioBtn2->Checked == true) {

if (txt2->Text == "" || txt2->Text == ",") {

RubToDol = 0;

}

else {

RubToDol = Convert::ToDouble(txt2->Text);

}

}

else if (txt2->Text != (Convert::ToString(EuroToRub)) && radioBtn3->Checked == true) {

if (txt2->Text == "" || txt2->Text == ",") {

EuroToRub = 0;

}

else {

EuroToRub = Convert::ToDouble(txt2->Text);

}

}

else if (txt2->Text != (Convert::ToString(RubToEuro)) && radioBtn4->Checked == true){

if (txt2->Text == "" || txt2->Text == ",") {

RubToEuro = 0;

}

else {

RubToEuro = Convert::ToDouble(txt2->Text);

}

}

}

//Обработчик нажатия на кнопку «Конвертировать»

private: System::Void btnAction\_Click(System::Object^ sender, System::EventArgs^ e) {

double count, value, result;

if ((txt1->Text == "," || txt1->Text == "" || txt1->Text == "0") || (txt2->Text == "," || txt2->Text == "" || txt2->Text == "0") || (txt1->Text->IndexOf(',') == 0 || txt2->Text->IndexOf(',') == 0)) {

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "Введите корректные значения в текстовые поля.", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Exclamation, MessageBoxDefaultButton::Button1);

}

else {

count = Convert::ToDouble(txt1->Text);

count = Math::Round(count, 2);

value = Convert::ToDouble(txt2->Text);

result = count \* value;

result = floor(result \* 100) / 100;

txt1->Text = Convert::ToString(count);

txtResult1->Text = Convert::ToString(count);

txtResult2->Text = Convert::ToString(result);

}

}

//Обработчик для выбора radioBtn1

private: System::Void radioBtn1\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

if (this->radioBtn1->Checked == true) {

txt2->Text = Convert::ToString(DolToRub);

this->lblExchangeRate->Text = L"Курс обмена ($/руб.):";

this->lblFrom->Text = L"$";

this->lblTo->Text = L"руб.";

}

else {

txtResult1->Text = L"";

txtResult2->Text = L"";

txt1->Text = L"";

}

}

//Обработчик для выбора radioBtn2

private: System::Void radioBtn2\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

if (this->radioBtn2->Checked == true) {

txt2->Text = Convert::ToString(RubToDol);

this->lblExchangeRate->Text = L"Курс обмена (руб./$):";

this->lblFrom->Text = L"р.";

this->lblTo->Text = L"$";

}

else {

txtResult1->Text = L"";

txtResult2->Text = L"";

txt1->Text = L"";

}

}

//Обработчик для выбора radioBtn3

private: System::Void radioBtn3\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

if (this->radioBtn3->Checked == true) {

txt2->Text = Convert::ToString(EuroToRub);

this->lblExchangeRate->Text = L"Курс обмена (€/руб.):";

this->lblFrom->Text = L"€";

this->lblTo->Text = L"руб.";

}

else {

txtResult1->Text = L"";

txtResult2->Text = L"";

txt1->Text = L"";

}

}

//Обработчик для выбора radioBtn4

private: System::Void radioBtn4\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

if (this->radioBtn4->Checked == true) {

txt2->Text = Convert::ToString(RubToEuro);

this->lblExchangeRate->Text = L"Курс обмена (руб./€):";

this->lblFrom->Text = L"р.";

this->lblTo->Text = L"€";

}

else {

txtResult1->Text = L"";

txtResult2->Text = L"";

txt1->Text = L"";

}

}

# **4 Результаты тестирования программы**

На рисунке 11 представлен интерфейс рабочей программы.

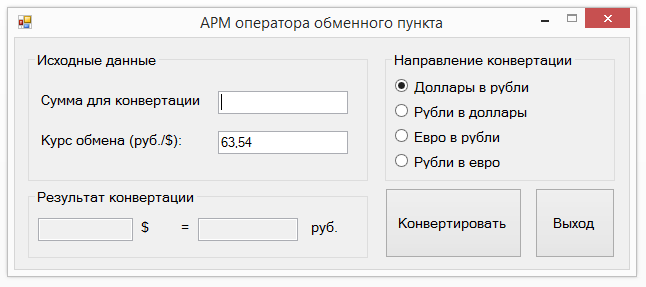


Рисунок 11 – Интерфейс программы

В таблице 8 представлены результаты тестирования программы.

Таблица 8 – Результаты тестирования программы

|  |  |
| --- | --- |
| **Введенные значения** | **Результат** |
| 1) Сумма для конвертации = 14, Доллары в рубли, курс обмена = 63,54 | 14$ = 889,56 руб. |
| 2) Сумма для конвертации = 148, Рубли в доллары, курс обмена = 0,016 | 148 руб. = 2,36 $ |
| 3) Сумма для конвертации = 120, Евро в рубли, курс обмена = 40,21 | 120 € = 4825,2 руб. |
| 4) Сумма для конвертации = 1884, Рубли в евро, курс обмена = 0,014 | 1884 руб. = 26,37€ |
| 5) Сумма для конвертации = пусто, Рубли в евро, курс обмена = 0,014 | Введите корректные значения в поля |
| 6) Сумма для конвертации = пусто, Рубли в доллары, курс обмена = пусто | Введите корректные значения в поля |

На рисунке 12 представлена корректная работа программы при пустых полях ввода конвертируемых значений.

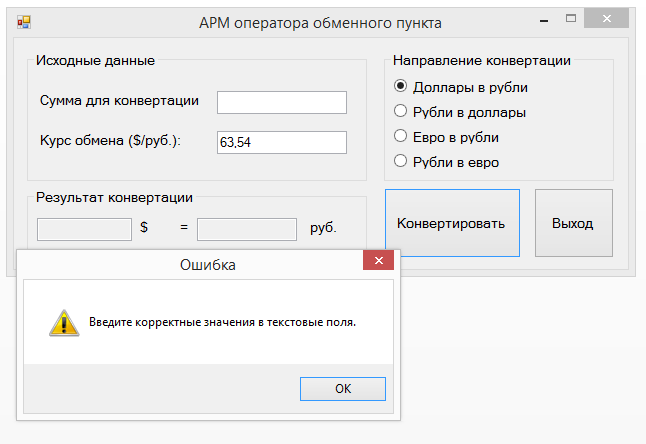


Рисунок 12 – Работа программы при некорректном вводе

На рисунке 13 представлена корректная работа программы с корректно введёнными данными.

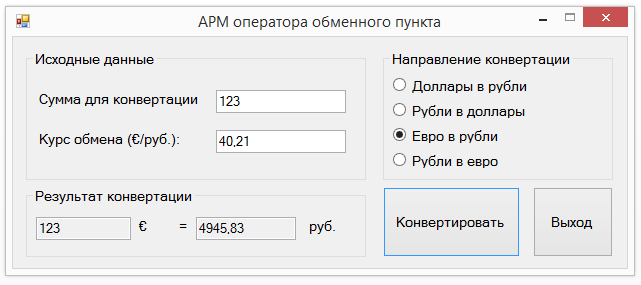


Рисунок 13 – Корректная работа программы

## **Вывод**

В ходе данной лабораторной работы были получены практические знания, связанные с алгоритмами ветвления, которые потребовались для создания простейшей программы, конвертирующей валюты. При этом были рассмотрены различные варианты возможной входной информации, предусмотрены некорректные значения вводимых данных.