МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №6

по дисциплине «Технологии программирования»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д. С./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М. Л./

Киров 2016

1. Задание на лабораторную работу

Необходимо разработать простой калькулятор вычисляющий сложение, умножение, деление и вычитание в различных системах счислениях. Базовым классом должна являться 10-я система счисления, она должна иметь 3 дочерних класса: 2-ую, 8-ую, 16-ую систему счисления.

1. Словесное описание алгоритма работы

При нажатии клавиши Enter считываем строку записанную в textbox. Разделяем данную строку на разделители «+», «-», «\*», «/». В результате получаем массив чисел в различных система счисления. Приводим все эти числа к десятичной системе счисления. Каждое число имеет свой префикс или постфикс обозначающий принадлежность к определенной системе счисления. Так например число в 16СС имеет префикс «0x», в 8СС – «0», двоичная система счисления имеет постфикс «b», а 10СС не имеет ни префикса, ни постфикса. Если какое-либо из чисел не удалось привести к 10СС, то выдается ошибка о неправильном формате команды. После приведения чисел происходит определение операции. Если строка имеет символ «+», то сложение, если «\*», то умножение и тд. После проведения арифметической операции необходимо привести число к нужной системе счисления. В зависимости от выбранного radiobutton’a выбирается необходимая система счисления.

1. Экранные формы

Экранные формы представлены на рисунках 1-3.

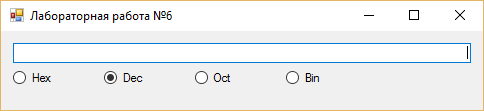


Рисунок 1 – Главное окно программы

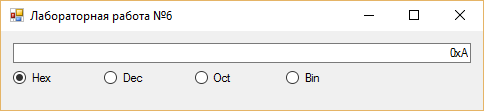


Рисунок 2 – Корректное поведение программы

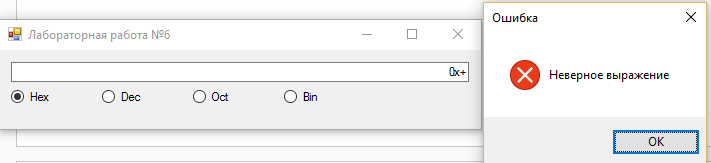


Рисунок 3 – Поведение программы при неверных данных

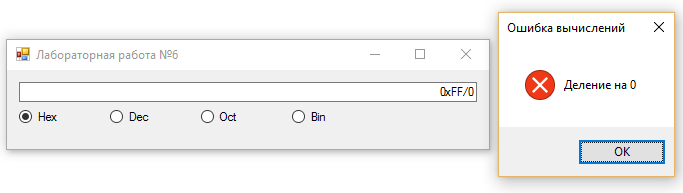


Рисунок 4 – Поведение программы при делении на 0

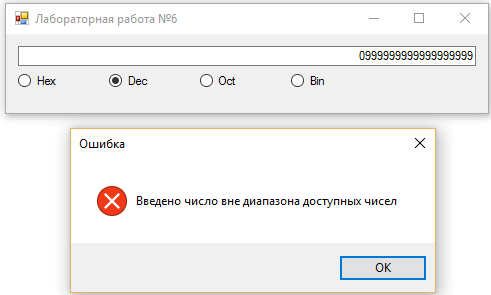


Рисунок 5 – Поведение программы при выходе за диапазон

1. Исходный код программы

Исходный код программы представлен на рисунке 6. Исходный код классов и вспомогательных функций представлен на рисунке 7

|  |
| --- |
| #pragma once  #include <set>  #include "nums.h"  namespace Lab6 {  char\* strToCharArr(System::String^ s) {  char \* c = new char[s->Length];  for (int i = 0; i < s->Length; i++)  {  c[i] = s[i];  }  return c;  }  int getLimit(System::String^ s) {  if (s->EndsWith("b")) {  return 32;  }  else if (s->StartsWith("0x")) {  return 10;  }  else if (s->StartsWith("0")) {  return 13;  }  else return 10;  }  using namespace System;  using namespace System::ComponentModel;  using namespace System::Collections;  using namespace System::Windows::Forms;  using namespace System::Data;  using namespace System::Drawing;  /// <summary>  /// Сводка для MainForm  /// </summary>  public ref class MainForm : public System::Windows::Forms::Form  {  public:  MainForm(void)  {  InitializeComponent();  //  //TODO: добавьте код конструктора  //  }  protected:  /// <summary>  /// Освободить все используемые ресурсы.  /// </summary>  ~MainForm()  {  if (components)  {  delete components;  }  }  private: System::Windows::Forms::TextBox^ textBox1;  private: System::Windows::Forms::RadioButton^ radioHex;  private: System::Windows::Forms::RadioButton^ radioDec;  private: System::Windows::Forms::RadioButton^ radioOct;  private: System::Windows::Forms::RadioButton^ radioBin;  protected:  private:  /// <summary>  /// Обязательная переменная конструктора.  /// </summary>  System::ComponentModel::Container ^components;  #pragma endregion  private: System::Void textBox1\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {  std::set<char> allowChar = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F',  'b', 'x', '+', '-', '\*', '/', 8};  array<wchar\_t>^ sep = { L'+', L'-', L'\*', L'/' };  if (!allowChar.count(e->KeyChar)) e->Handled = true;  if ( (e->KeyChar != 13) && (e->KeyChar != '+') && (e->KeyChar != '-') && (e->KeyChar != '\*') && (e->KeyChar != '/') ) return;  array<System::String^>^ nums = textBox1->Text->Split(sep);  if (nums->Length < 2) return;  Dec a, b, c;  try  {  if ((nums[0]->Length > 8) || (nums[1]->Length > 8)) throw BadFormat();  a = strToNum(nums[0]);  b = strToNum(nums[1]);  }  catch (BadFormat)  {  MessageBox::Show("Неверное выражение", "Ошибка", System::Windows::Forms::MessageBoxButtons::OK, System::Windows::Forms::MessageBoxIcon::Error);  e->Handled = true;  return;  }  if (textBox1->Text->Contains("+")) {  c = a + b;  }  else if (textBox1->Text->Contains("-")) {  c = a - b;  }  else if (textBox1->Text->Contains("\*")) {  c = a \* b;  }  else {  try {  c = a / b;  }  catch (DivideByZero) {  MessageBox::Show("Деление на 0", "Ошибка вычислений", System::Windows::Forms::MessageBoxButtons::OK, System::Windows::Forms::MessageBoxIcon::Error);  textBox1->Clear();  return;  }  }  if (radioHex->Checked) {  textBox1->Text = Hex(c.getVal()).show();  }  else if (radioDec->Checked) {  textBox1->Text = c.show();  }  else if (radioOct->Checked) {  textBox1->Text = Oct(c.getVal()).show();  }  else {  textBox1->Text = Bin(c.getVal()).show();  }  }  private: System::Void radioHex\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {  int limit = getLimit(textBox1->Text);  if (textBox1->Text->Length > limit) {  MessageBox::Show("Введено число вне диапазона доступных чисел", "Ошибка", System::Windows::Forms::MessageBoxButtons::OK, System::Windows::Forms::MessageBoxIcon::Error);  return;  }  char\* c = strToCharArr(textBox1->Text);  Hex h;  if (textBox1->Text->EndsWith("b")) {  h = Hex(strtol(c, NULL, 2));  }  else {  h = Hex(strtol(c, NULL, 0));  }  textBox1->Text = h.show();  }  private: System::Void radioDec\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {  int limit = getLimit(textBox1->Text);  if (textBox1->Text->Length > limit) {  MessageBox::Show("Введено число вне диапазонаа доступных чисел", "Ошибка", System::Windows::Forms::MessageBoxButtons::OK, System::Windows::Forms::MessageBoxIcon::Error);  return;  }    char\* c = strToCharArr(textBox1->Text);  Dec h;  if (textBox1->Text->EndsWith("b")) {  h = Dec(strtol(c, NULL, 2));  }  else {  h = Dec(strtol(c, NULL, 0));  }  textBox1->Text = h.show();  }  private: System::Void radioOct\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {  int limit = getLimit(textBox1->Text);  if (textBox1->Text->Length > limit) {  MessageBox::Show("Введено число вне диапазона доступных чисел", "Ошибка", System::Windows::Forms::MessageBoxButtons::OK, System::Windows::Forms::MessageBoxIcon::Error);  return;  }  char\* c = strToCharArr(textBox1->Text);  Oct h;  if (textBox1->Text->EndsWith("b")) {  h = Oct(strtol(c, NULL, 2));  }  else {  h = Oct(strtol(c, NULL, 0));  }  textBox1->Text = h.show();  }  private: System::Void radioBin\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {  int limit = getLimit(textBox1->Text);  if (textBox1->Text->Length > limit) {  MessageBox::Show("Введено число вне диапазона доступных чисел", "Ошибка", System::Windows::Forms::MessageBoxButtons::OK, System::Windows::Forms::MessageBoxIcon::Error);  return;  }    char\* c = strToCharArr(textBox1->Text);  Bin h;  if (textBox1->Text->EndsWith("b")) {  return;  }  else {  h = Bin(strtol(c, NULL, 0));  }  textBox1->Text = h.show();  }  };  } |

Рисунок 6 – Исходный код программы

|  |
| --- |
| #pragma once  #include <cstdlib>  #include <iostream>  class DivideByZero {};  class BadFormat {};  class Dec  {  public:  Dec(long int n) : val(n) {};  Dec(System::String^ value) {  char \* c = new char[value->Length];  for (int i = 0; i < value->Length; i++)  {  c[i] = value[i];  }  val = strtol(c, NULL, 10);  if ((val == 0) && (value != "0")) throw BadFormat();  }  Dec() {};  ~Dec() {};  int getVal() { return val; }  Dec operator +(Dec b) {  Dec r = Dec(val + b.getVal());  return r;  }  Dec operator -(Dec b) {  Dec r = Dec(val - b.getVal());  return r;  }  Dec operator \*(Dec b) {  Dec r = Dec(val \* b.getVal());  return r;  }  Dec operator /(Dec b) {  if (b.getVal() == 0) throw DivideByZero();  Dec r = Dec(val / b.getVal());  return r;  }  System::String^ show() {  System::String^ res;  char\* cNum = new char[32];  itoa(val, cNum, 10);  for (int i = 0; cNum[i] != '\0'; i++)  {  res += wchar\_t(cNum[i]);  }  return res;  }  protected:  long int val;  };  class Hex : public Dec  {  public:  Hex() {};  Hex(long int value) {  val = value;  }  Hex(System::String^ value) {  char \* c = new char[value->Length];  for (int i = 0; i < value->Length; i++)  {  c[i] = value[i];  }  val = strtol(c, NULL, 16);  if ((val == 0) && (value != "0")) throw BadFormat();  }  ~Hex() {};  System::String^ show() {  System::String^ res;  char\* cNum = new char[32];  itoa(val, cNum, 16);  for (int i = 0; cNum[i] != '\0'; i++)  {  res += wchar\_t(cNum[i]);  }  return "0x" + res->ToUpper();  }  };  class Oct : public Dec  {  public:  Oct() {};  Oct(long int value) {  val = value;  }  Oct(System::String^ value) {  char \* c = new char[value->Length];  for (int i = 0; i < value->Length; i++)  {  c[i] = value[i];  }  val = strtol(c, NULL, 8);  if ((val == 0) && (value != "0")) throw BadFormat();  }  ~Oct() {};  System::String^ show() {  System::String^ res;  char\* cNum = new char[32];  itoa(val, cNum, 8);  for (int i = 0; cNum[i] != '\0'; i++)  {  res += wchar\_t(cNum[i]);  }  return "0" + res;  }  };  class Bin : public Dec  {  public:  Bin() {};  Bin(long int value) {  val = value;  }  Bin(System::String^ value) {  char \* c = new char[value->Length];  for (int i = 0; i < value->Length; i++)  {  c[i] = value[i];  }  val = strtol(c, NULL, 2);  if ((val == 0) && (value != "0")) throw BadFormat();  }  ~Bin() {};  System::String^ show() {  System::String^ res;  char\* cNum = new char[32];  itoa(val, cNum, 2);  for (int i = 0; cNum[i] != '\0'; i++)  {  res += wchar\_t(cNum[i]);  }  return res + "b";  }  };  Dec strToNum(System::String^ s) {  if (s->StartsWith("0x")) {  return (Dec)Hex(s);  }  else if (s->StartsWith("0")) {  return (Dec)Oct(s);  }  else if (s->EndsWith("b")) {  return (Dec)Bin(s);  }  else return Dec(s);  } |

Рисунок 7 – Классы и вспомогательные функции

1. Вывод

В ходе выполнения лабораторной работы были полученные необходимые знания базового синтаксиса языка программирования С++. Было изучено объектно-ориентированное программирование и на основе полученных знаний был создан калькулятор позволяющий вычислять простые выражения в различных системах счисления. Данные знания являются фундаментальными и необходимы для дальнейшего продолжения изучения различных технологий программирования и языка С++.