Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное   
образовательное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Лабораторная Работа №3. Файлы

Отчет по лабораторной работе №3 по учебной дисциплине «Алгоритмические языки и программирование»

по специальности 09.03.01 Информатика и вычислительная техника

|  |  |
| --- | --- |
|  | Руководитель  / И. Ю. Кулаков  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |
|  | Студент группы 3091  / А.Н. Архангельская  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

1. Постановка задачи

Разработка программы с графическим интерфейсом для работы с файлами.

* 1. Наименование задачи – Лабораторная работа №3. Файлы.
  2. Словесное описание:

Реализовать и разработать программу с графическим интерфейсом для работы с файлами.

| Объект программы | Имя этого объекта в программе | Тип данных | Диапазоны представления | Простая/  структура | Вход/  выход/  Константа |
| --- | --- | --- | --- | --- | --- |
| Имя текстового  файла | filename | String^ | 0-225 | Простая | Вход |
| Имя текстового  файла | creatfilename | String^ | 0-225 | Простая | Выход |

Внешние спецификации данных

* 1. Внешние спецификации функций программы

Программа должна редактировать выбранный файл по выбранному редактированию. Выводить текст файла. Должна быть реализовано сохранение и создание файлов.

* 1. Спецификации интерфейса

Для написания графического интерфейса была использована библиотека Windows Forms.

Плюсы интерфейса заключаются в том, что реализован добавление в стек любого числового типа.

Реализована обработка повторного нажатия ключевых кнопок, что предотвращает некорректный вывод данных.  
В приложении реализована работа c данными сортировка их , и математические действия над ними.

* 1. Внешние данные тестирования

Таблица 1. Тест-кейс №1, выход из приложения

| Идентификатор | | | T01 |
| --- | --- | --- | --- |
| Заголовок | | | Проверка выхода из приложения |
| Предусловие | | | Приложение не запущено |
| Ожидаемый результат | | | Успешный выход из приложения |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Запуск Lab6.exe | Открытие главного окна приложения | Pass |
| 2 | Нажатие на кнопку «Выход» | Выход из приложения | Pass |
| Результат | | | Pass |

Таблица 2. Тест-кейс №2, кнопка "Редактирование"

| Идентификатор | | | T02 |
| --- | --- | --- | --- |
| Заголовок | | | Проверка работоспособности кнопки «Редактирование» |
| Предусловие | | | Приложение запущено. |
| Ожидаемый результат | | | В тексте убрались все пробелы. |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Запуск lab3\_f.exe | Открытие главного окна приложения | Pass |
| 2 | Нажатие кнопки «Открыть файл» | Появиться диалоговое окно | Pass |
| 3 | Выбрать способ редактирования «Удаление всех пробелов в тексте» |  | Pass |
| 4 | Нажатие кнопки «Вывести текст» | Открылся графа с текстом из файла | Pass |
| 5 | Нажатие кнопки «Редактировать» | В тексте убрались все пробелы. | Pass |
| Результат | | | Pass |

Таблица 3. Тест-кейс №3, кнопка "Создать файл"

| Идентификатор | | | T03 |
| --- | --- | --- | --- |
| Заголовок | | | Проверка работоспособности кнопки «Создать файл» |
| Предусловие | | | Приложение запущено |
| Ожидаемый результат | | | Создаться файл |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Запуск lab3\_f.exe | Открытие главного окна приложения | Pass |
| 2 | Нажатие кнопки «Создать файл» | Появиться диалоговое окно | Pass |
| 3 | Ввести название нового файла |  | Pass |
| 4 | Нажать кнопку «Сохранить» | Файл с введённым названием создаться | Pass |
| Результат | | | Pass |

Таблица 4. Тест-кейс №4, контроль ввода

| Идентификатор | | | T04 |
| --- | --- | --- | --- |
| Заголовок | | | Проверка контроля ввода |
| Предусловие | | | Приложение запущена |
| Ожидаемый результат | | | Выведена ошибка |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Запуск lab3\_f.exe | Открытие главного окна приложения | Pass |
| 2 | Нажатие кнопки «Открыть файл» | Появиться диалоговое окно | Pass |
| 3 | Выбор файлом с расширением .jpg | Выведена ошибка | Pass |
| Результат | | | Pass |

* 1. Пример работающей программы

Чтобы добавить элемент в стек, введите число в текстовое поле с надписью "Введите число" и нажмите кнопку "Добавить". Введенное число будет добавлено в стек и отображено в списке элементов стека. Для удаления элемента из стека нажмите кнопку "Удалить". Верхний элемент стека будет удален, и список элементов стека обновится. Для выполнения операций выберите операцию из выпадающего списка (например, "Сумма", "Разность", "Умножение", "Деление"), введите число для выполнения операции и нажмите кнопку "Выполнить". Результат операции будет отображен, и стек обновится. Чтобы отсортировать стек, нажмите кнопку "Сортировка".

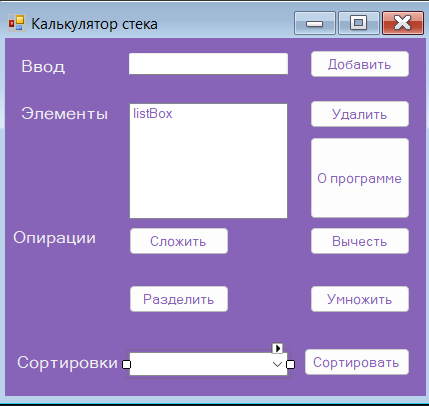


Рисунок 1. Интерфейс приложения

1. Проектирование задачи

Реализовать приложение с графическим интерфейсом, использую библиотеку Windows Forms, для работы с разными типами данных.

* 1. Наименование программы

Lab6.exe

* 1. Уточненное словесное описание задачи

Необходимо реализовать программу, в которой будет реализован класс стека, который может хранить элементы любого типа данных и выполнение любых математических операций над ними.

* 1. Выбор метода решения поставленной задачи

Работа алгоритма осуществляется путём ввода числа в стек, реализованный в отдельном файле.

После нажатия кнопки «Добавить» число из строки ввода читается и из ТД String^ переводиться в ТД double и добавляется в стек.

После нажатия кнопок «Сложить» «Вычесть» «Разделить» «Умножить» вырезается верхние значение выполняется математические действия в соответствии с кнопкой, потом вставляются в запасной стек. А потом уже вырезаются из запасного и вставляются в основной

После выбора сортировки и нажатии кнопки «Сортировать» числа в стеке сортируются по выбранной сортировки.

Кнопки «О программе» выводят на экран всплывающее окно, в котором описан алгоритм работы с программой в данной области.

* 1. Уточненные глобальные данные программы и пользовательские типы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект программы | Имя этого объекта в программе | Характеристики | | | Как используется в программе |
| Тип данных | Диапазоны представления | Прост/структура/ константа |
| MyForm.h | stack | Stack<double>\* | double | Структура | Стек основной |
| MyForm.h | stack | Stack<double>\* | double | Структура | Стек вспомогательный |

* 1. Декомпозиция функций

Описывается каждая функция и процедура в программе

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Назначение | Имя | Параметры | | Процедура/ функция | Тестируется/  не тестируется |
| Вход:  тип | Выход: тип |
| Добавление элементов в стек | addButton\_Click | System::Object^, System::EventArgs^ | Процедура | Процедура | Нет |
| Удаление элементов из стека | removeButton\_Click | System::Object^, System::EventArgs^ | Процедура | System::Void | Нет |
| Выполнение операции сложения | addOpButton\_Click | System::Object^, System::EventArgs^ | Процедура | System::Void | Нет |
| Выполнение операции вычитания | subOpButton\_Click | System::Object^, System::EventArgs^ | Процедура | System::Void | Нет |
| Выполнение операции умножения | mulOpButton\_Click | System::Object^, System::EventArgs^ | Процедура | System::Void | Нет |
| Выполнение операции деления | divOpButton\_Click | System::Object^, System::EventArgs^ | Процедура | System::Void | Нет |
| Отображение информации о программе | aboutButton\_Click | System::Object^, System::EventArgs^ | Процедура | System::Void | Нет |
| Выполнение операции сортировки | button1\_Click | System::Object^, System::EventArgs^ | Процедура | System::Void | Нет |
| Вспомогательная функция выполнения операций | PerformOperation | System::Object^, System::EventArgs^ | Процедура | System::Void | Нет |

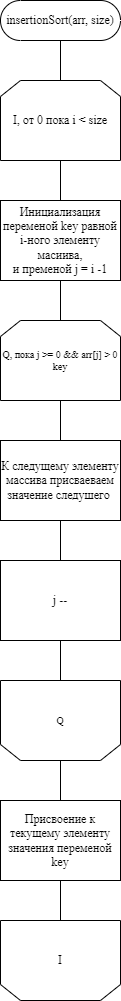
* 1. Блок-схема

Блок-схемы функций в Приложении А.

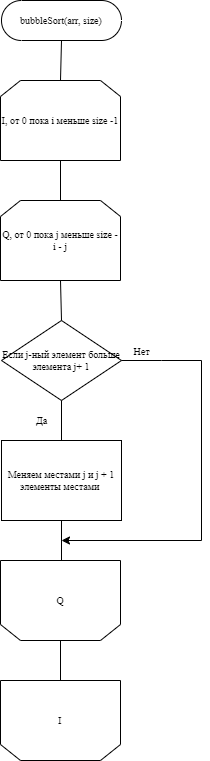
* 1. Алгоритмизация

Листинг кода программы в Приложение Б.

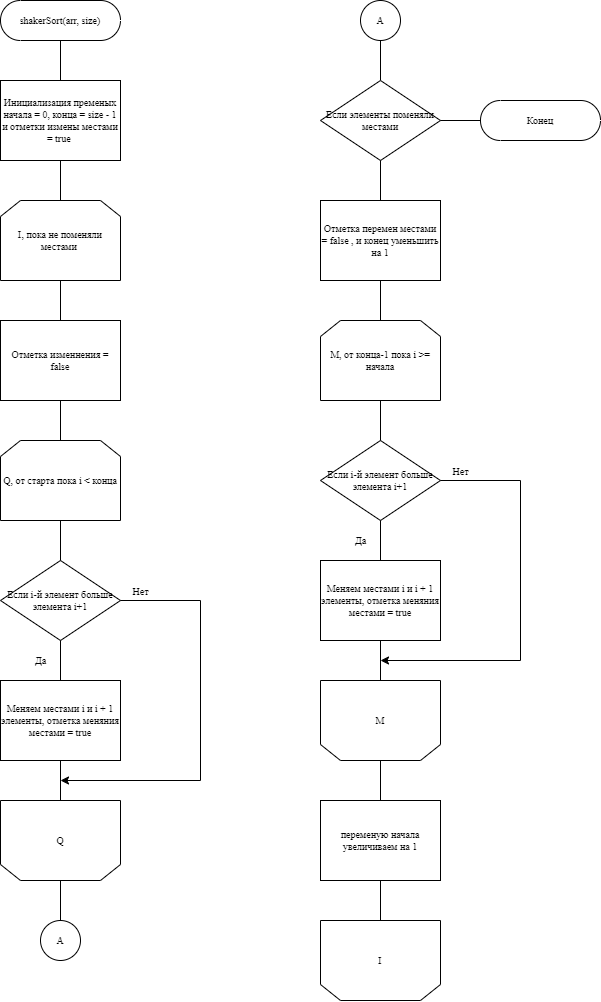
Приложении A



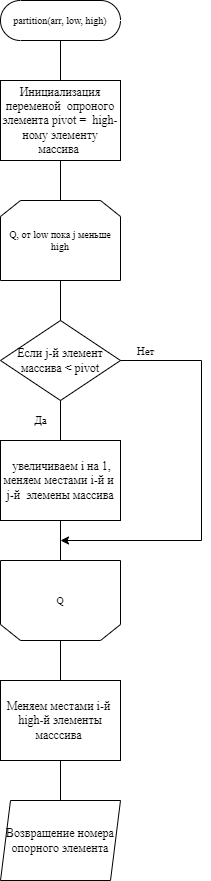
Рисунок



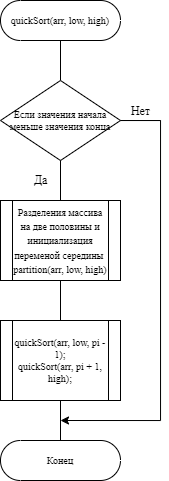
Рисунок



Рисунок



Рисунок



Рисунок

Приложение Б.1

Листинг программы, файл MyForm.h

#pragma once

#include "Stack.h"

namespace Lab6 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

public ref class MyForm : public System::Windows::Forms::Form {

public:

MyForm(void) {

InitializeComponent();

stack = new ::Stack<double>();

stack2 = new ::Stack<double>();// Явное указание на глобальное пространство имен

}

protected:

~MyForm() {

this->!MyForm(); // Вызов финализатора (деструктора) управляемого ресурса

if (components) {

delete components;

}

}

// Финализатор (деструктор) управляемого ресурса

!MyForm() {

if (stack) {

delete stack;

stack = nullptr;

}

}

private:

::Stack<double>\* stack; // Используем явное глобальное пространство имен

::Stack<double>\* stack2;

System::ComponentModel::Container^ components;

System::Windows::Forms::TextBox^ inputBox;

System::Windows::Forms::ListBox^ listBox;

System::Windows::Forms::Button^ addButton;

System::Windows::Forms::Button^ removeButton;

System::Windows::Forms::Button^ addOpButton;

System::Windows::Forms::Button^ subOpButton;

System::Windows::Forms::Button^ mulOpButton;

System::Windows::Forms::Button^ divOpButton;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::ComboBox^ sortedBox;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::Button^ button1;

System::Windows::Forms::Button^ aboutButton;

void InitializeComponent(void) {

this->inputBox = (gcnew System::Windows::Forms::TextBox());

this->listBox = (gcnew System::Windows::Forms::ListBox());

this->addButton = (gcnew System::Windows::Forms::Button());

this->removeButton = (gcnew System::Windows::Forms::Button());

this->addOpButton = (gcnew System::Windows::Forms::Button());

this->subOpButton = (gcnew System::Windows::Forms::Button());

this->mulOpButton = (gcnew System::Windows::Forms::Button());

this->divOpButton = (gcnew System::Windows::Forms::Button());

this->aboutButton = (gcnew System::Windows::Forms::Button());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->sortedBox = (gcnew System::Windows::Forms::ComboBox());

this->label4 = (gcnew System::Windows::Forms::Label());

this->button1 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// inputBox

//

this->inputBox->ForeColor = System::Drawing::SystemColors::Desktop;

this->inputBox->Location = System::Drawing::Point(124, 15);

this->inputBox->Margin = System::Windows::Forms::Padding(4);

this->inputBox->Name = L"inputBox";

this->inputBox->Size = System::Drawing::Size(159, 22);

this->inputBox->TabIndex = 0;

//

// listBox

//

this->listBox->ForeColor = System::Drawing::SystemColors::Desktop;

this->listBox->FormattingEnabled = true;

this->listBox->ItemHeight = 16;

this->listBox->Location = System::Drawing::Point(124, 65);

this->listBox->Margin = System::Windows::Forms::Padding(4);

this->listBox->Name = L"listBox";

this->listBox->Size = System::Drawing::Size(159, 116);

this->listBox->TabIndex = 1;

//

// addButton

//

this->addButton->ForeColor = System::Drawing::SystemColors::Desktop;

this->addButton->Location = System::Drawing::Point(305, 12);

this->addButton->Margin = System::Windows::Forms::Padding(4);

this->addButton->Name = L"addButton";

this->addButton->Size = System::Drawing::Size(100, 28);

this->addButton->TabIndex = 2;

this->addButton->Text = L"Добавить";

this->addButton->UseVisualStyleBackColor = true;

this->addButton->Click += gcnew System::EventHandler(this, &MyForm::addButton\_Click);

//

// removeButton

//

this->removeButton->ForeColor = System::Drawing::SystemColors::Desktop;

this->removeButton->Location = System::Drawing::Point(305, 62);

this->removeButton->Margin = System::Windows::Forms::Padding(4);

this->removeButton->Name = L"removeButton";

this->removeButton->Size = System::Drawing::Size(100, 28);

this->removeButton->TabIndex = 3;

this->removeButton->Text = L"Удалить";

this->removeButton->UseVisualStyleBackColor = true;

this->removeButton->Click += gcnew System::EventHandler(this, &MyForm::removeButton\_Click);

//

// addOpButton

//

this->addOpButton->ForeColor = System::Drawing::SystemColors::Desktop;

this->addOpButton->Location = System::Drawing::Point(124, 189);

this->addOpButton->Margin = System::Windows::Forms::Padding(4);

this->addOpButton->Name = L"addOpButton";

this->addOpButton->Size = System::Drawing::Size(100, 28);

this->addOpButton->TabIndex = 4;

this->addOpButton->Text = L"Сложить";

this->addOpButton->UseVisualStyleBackColor = true;

this->addOpButton->Click += gcnew System::EventHandler(this, &MyForm::addOpButton\_Click);

//

// subOpButton

//

this->subOpButton->ForeColor = System::Drawing::SystemColors::Desktop;

this->subOpButton->Location = System::Drawing::Point(305, 189);

this->subOpButton->Margin = System::Windows::Forms::Padding(4);

this->subOpButton->Name = L"subOpButton";

this->subOpButton->Size = System::Drawing::Size(100, 28);

this->subOpButton->TabIndex = 5;

this->subOpButton->Text = L"Вычесть";

this->subOpButton->UseVisualStyleBackColor = true;

this->subOpButton->Click += gcnew System::EventHandler(this, &MyForm::subOpButton\_Click);

//

// mulOpButton

//

this->mulOpButton->ForeColor = System::Drawing::SystemColors::Desktop;

this->mulOpButton->Location = System::Drawing::Point(305, 247);

this->mulOpButton->Margin = System::Windows::Forms::Padding(4);

this->mulOpButton->Name = L"mulOpButton";

this->mulOpButton->Size = System::Drawing::Size(100, 28);

this->mulOpButton->TabIndex = 6;

this->mulOpButton->Text = L"Умножить";

this->mulOpButton->UseVisualStyleBackColor = true;

this->mulOpButton->Click += gcnew System::EventHandler(this, &MyForm::mulOpButton\_Click);

//

// divOpButton

//

this->divOpButton->ForeColor = System::Drawing::SystemColors::Desktop;

this->divOpButton->Location = System::Drawing::Point(124, 247);

this->divOpButton->Margin = System::Windows::Forms::Padding(4);

this->divOpButton->Name = L"divOpButton";

this->divOpButton->Size = System::Drawing::Size(100, 28);

this->divOpButton->TabIndex = 7;

this->divOpButton->Text = L"Разделить";

this->divOpButton->UseVisualStyleBackColor = true;

this->divOpButton->Click += gcnew System::EventHandler(this, &MyForm::divOpButton\_Click);

//

// aboutButton

//

this->aboutButton->ForeColor = System::Drawing::SystemColors::Desktop;

this->aboutButton->Location = System::Drawing::Point(305, 99);

this->aboutButton->Margin = System::Windows::Forms::Padding(4);

this->aboutButton->Name = L"aboutButton";

this->aboutButton->Size = System::Drawing::Size(100, 82);

this->aboutButton->TabIndex = 8;

this->aboutButton->Text = L"О программе";

this->aboutButton->UseVisualStyleBackColor = true;

this->aboutButton->Click += gcnew System::EventHandler(this, &MyForm::aboutButton\_Click);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->ForeColor = System::Drawing::SystemColors::ButtonFace;

this->label1->Location = System::Drawing::Point(12, 18);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(53, 20);

this->label1->TabIndex = 9;

this->label1->Text = L"Ввод";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->ForeColor = System::Drawing::SystemColors::ButtonFace;

this->label2->Location = System::Drawing::Point(12, 65);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(96, 20);

this->label2->TabIndex = 10;

this->label2->Text = L"Элементы";

//

// label3

//

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label3->ForeColor = System::Drawing::SystemColors::ButtonFace;

this->label3->Location = System::Drawing::Point(4, 189);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(92, 20);

this->label3->TabIndex = 11;

this->label3->Text = L"Опирации";

//

// sortedBox

//

this->sortedBox->ForeColor = System::Drawing::SystemColors::Desktop;

this->sortedBox->FormattingEnabled = true;

this->sortedBox->Items->AddRange(gcnew cli::array< System::Object^ >(5) {

L"a.Поразрядная;", L"b.Вставками;", L"c.Пузырьковая;",

L"d.Шейкера;", L"e.Быстрая."

});

this->sortedBox->Location = System::Drawing::Point(124, 314);

this->sortedBox->Name = L"sortedBox";

this->sortedBox->Size = System::Drawing::Size(159, 24);

this->sortedBox->TabIndex = 12;

//

// label4

//

this->label4->AutoSize = true;

this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label4->ForeColor = System::Drawing::SystemColors::ButtonFace;

this->label4->Location = System::Drawing::Point(8, 314);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(110, 20);

this->label4->TabIndex = 13;

this->label4->Text = L"Сортировки";

//

// button1

//

this->button1->ForeColor = System::Drawing::SystemColors::Desktop;

this->button1->Location = System::Drawing::Point(299, 310);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(106, 28);

this->button1->TabIndex = 14;

this->button1->Text = L"Сортировать";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackColor = System::Drawing::SystemColors::Desktop;

this->ClientSize = System::Drawing::Size(421, 358);

this->Controls->Add(this->button1);

this->Controls->Add(this->label4);

this->Controls->Add(this->sortedBox);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->aboutButton);

this->Controls->Add(this->divOpButton);

this->Controls->Add(this->mulOpButton);

this->Controls->Add(this->subOpButton);

this->Controls->Add(this->addOpButton);

this->Controls->Add(this->removeButton);

this->Controls->Add(this->addButton);

this->Controls->Add(this->listBox);

this->Controls->Add(this->inputBox);

this->Margin = System::Windows::Forms::Padding(4);

this->Name = L"MyForm";

this->Text = L"Калькулятор стека";

this->ResumeLayout(false);

this->PerformLayout();

}

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

double value;

if (Double::TryParse(inputBox->Text, value)) {

stack->push(value);

listBox->Items->Add(value);

inputBox->Clear();

}

else {

MessageBox::Show("Введите корректное число", "Ошибка ввода", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

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

if (!stack->isEmpty()) {

stack->pop();

listBox->Items->RemoveAt(listBox->Items->Count - 1);

}

else {

MessageBox::Show("Стек пуст", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

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

PerformOperation(Operations::Add);

}

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

PerformOperation(Operations::Subtract);

}

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

PerformOperation(Operations::Multiply);

}

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

PerformOperation(Operations::Divide);

}

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

MessageBox::Show("Калькулятор стека\nВерсия 1.0\nАвтор: Ваше имя", "О программе", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

enum class Operations {

Add,

Subtract,

Multiply,

Divide

};

void PerformOperation(Operations operation) {

double operand;

if (Double::TryParse(inputBox->Text, operand)) {

if (!stack->isEmpty()) {

switch (operation) {

case Operations::Add:

while (!stack->isEmpty()) {

double result;

double top = stack->top();

result = top + operand;

stack2->push(result);

stack->pop(); // Удаляем элемент из первого стека

}

// Очищаем listBox

listBox->Items->Clear();

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

break;

case Operations::Subtract:

while (!stack->isEmpty()) {

double result;

double top = stack->top();

result = top - operand;

stack2->push(result);

stack->pop(); // Удаляем элемент из первого стека

}

// Очищаем listBox

listBox->Items->Clear();

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

break;

case Operations::Multiply:

while (!stack->isEmpty()) {

double result;

double top = stack->top();

result = top \* operand;

stack2->push(result);

stack->pop(); // Удаляем элемент из первого стека

}

// Очищаем listBox

listBox->Items->Clear();

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

break;

case Operations::Divide:

if (operand != 0) {

while (!stack->isEmpty()) {

double result;

double top = stack->top();

result = top / operand;

stack2->push(result);

stack->pop(); // Удаляем элемент из первого стека

}

// Очищаем listBox

listBox->Items->Clear();

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

break;

}

else {

MessageBox::Show("Деление на ноль невозможно", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

break;

}

}

else {

MessageBox::Show("Стек пуст", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

else {

MessageBox::Show("Введите корректное число", "Ошибка ввода", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

if (this->sortedBox->SelectedItem == "a.Поразрядная;") {

listBox->Items->Clear();

stack->radixSort();

while (!stack->isEmpty()) {

double top = stack->top();

stack2->push(top);

stack->pop(); // Удаляем элемент из первого стека

}

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

}

else if (this->sortedBox->SelectedItem == "b.Вставками;") {

listBox->Items->Clear();

stack->insertionSort();

while (!stack->isEmpty()) {

double top = stack->top();

stack2->push(top);

stack->pop(); // Удаляем элемент из первого стека

}

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

}

else if (this->sortedBox->SelectedItem == "c.Пузырьковая;") {

listBox->Items->Clear();

stack->bubbleSort();

while (!stack->isEmpty()) {

double top = stack->top();

stack2->push(top);

stack->pop(); // Удаляем элемент из первого стека

}

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

}

else if (this->sortedBox->SelectedItem == "d.Шейкера;") {

listBox->Items->Clear();

stack->cocktailShakerSort();

while (!stack->isEmpty()) {

double top = stack->top();

stack2->push(top);

stack->pop(); // Удаляем элемент из первого стека

}

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

}

else if (this->sortedBox->SelectedItem == "e.Быстрая.") {

listBox->Items->Clear();

stack->radixSort();

while (!stack->isEmpty()) {

double top = stack->top();

stack2->push(top);

stack->pop(); // Удаляем элемент из первого стека

}

// Переносим элементы обратно из второго стека в первый и обновляем listBox

while (!stack2->isEmpty()) {

double top = stack2->top();

stack->push(top);

listBox->Items->Add(top);

stack2->pop(); // Удаляем элемент из второго стека

}

}

else {

MessageBox::Show(this, "Выберете сортировку", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

}

};

}

Приложение Б.2

Листинг программы, файл MyForm.cpp

#include "MyForm.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

void main(array<String^>^ args) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Lab6::MyForm form;

Application::Run(% form);

}

Листинг программы, файл .h

#include <type\_traits> // For std::is\_integral

#pragma once

#include <vector>

#include <stdexcept>

#include <algorithm> // For std::max\_element

#include <cmath> // For std::pow

template<typename T>

class Stack {

public:

void push(T value) {

data.push\_back(value);

}

T pop() {

if (data.empty()) {

throw std::out\_of\_range("Stack is empty");

}

T value = data.back();

data.pop\_back();

return value;

}

T top() const {

if (data.empty()) {

throw std::out\_of\_range("Stack is empty");

}

return data.back();

}

bool isEmpty() const {

return data.empty();

}

size\_t size() const {

return data.size();

}

void insertionSort() {

for (size\_t i = 1; i < data.size(); i++) {

T key = data[i];

int j = i - 1;

while (j >= 0 && data[j] > key) {

data[j + 1] = data[j];

j--;

}

data[j + 1] = key;

}

}

void countingSort(int exp) {

int n = data.size();

std::vector<T> output(n);

std::vector<int> count(10, 0);

for (int i = 0; i < n; i++) {

count[(data[i] / exp) % 10]++;

}

for (int i = 1; i < 10; i++) {

count[i] += count[i - 1];

}

for (int i = n - 1; i >= 0; i--) {

output[count[(data[i] / exp) % 10] - 1] = data[i];

count[(data[i] / exp) % 10]--;

}

for (int i = 0; i < n; i++) {

data[i] = output[i];

}

}

void bubbleSort() {

if (data.empty())

return;

int n = data.size();

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (data[j] > data[j + 1]) {

std::swap(data[j], data[j + 1]);

}

}

}

}

void cocktailShakerSort() {

if (data.empty())

return;

int left = 0;

int right = data.size() - 1;

while (left <= right) {

for (int i = left; i < right; i++) {

if (data[i] > data[i + 1]) {

std::swap(data[i], data[i + 1]);

}

}

right--;

for (int i = right; i > left; i--) {

if (data[i] < data[i - 1]) {

std::swap(data[i], data[i - 1]);

}

}

left++;

}

}

void quickSort(int low, int high) {

if (low < high) {

int pi = partition(low, high);

quickSort(low, pi - 1);

quickSort(pi + 1, high);

}

}

int partition(int low, int high) {

T pivot = data[high];

int i = (low - 1);

for (int j = low; j <= high - 1; j++) {

if (data[j] < pivot) {

i++;

std::swap(data[i], data[j]);

}

}

std::swap(data[i + 1], data[high]);

return (i + 1);

}

private:

std::vector<T> data;

};