Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУЖЕНИХ ТА

ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

ВИКОНАВ

студент академічної групи

КБ-21

Горденка А.В.

ПЕРЕВІРИВ

викладач кафедри

кібербезпеки

та програмного забезпечення

Усік П.С

Кропивницький – 2022

**Лабораторна робота №9**

**Тема:**Реалзіація програмних модулів розгалужених та інтераційних обчислювалльних процесів

**Мета** полягає у набутті ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів С/С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

**Завдання**

1.Реалізувати функції розв’язування задач 9.1–9.3 як складових статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище, створений під час виконання лабораторної роботи №8).

2.Реалізувати програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesПрізвище.а.

**Варіант № 13**



Рисунок 1 – Умова задачі 9.1

**Алгоритмізація програмного модуля задачі 9.1**

Вхід: Користувач вводить силу вiтру.

Вихід: В результаті отримуємо бал Бофота та опис дii



Рисунок 2 – Умова задачі 9.2

**Алгоритмізація програмного модуля задачі 9.2**

Вхід: температура за шкалою цельсiя.

Вихід: Середньомiсячна температура в цельсiях та фаренгейтах.



Рисунок 3 – Умова задачі 9.3

**Алгоритмізація програмного модуля задачі 9.3**

Вхід: користувач вводить натуральне число N від 0 до 10008000.

Вихід: якщо біт D14 числа N рівний 1, кількість двійкових нулів у ньому, інакше - кiлькысть двійкових одиниць.



Рисунок 4 – Умова задачі 9.4

**Алгоритмізація програмного модуля задачі 9.4**

Якщо користувач вводить символ “7” викликається функція s\_calculation (). Якщо користувач вводить символ “ 5” – викликається функція задачі 9.1. Якщо вводять символ “4” – функція задачі 9.2. “3” – функція задачі 9.3, вводимо натуральне число;якщо користувач вводить інші символи,вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “v”, “V” або “A”, відбувається вихід з програми, інакше - виконання програми повторюється.

**Висновок:** Мета лабораторної роботи № 9 з дисципліни “Базові методології та технології програмування на тему “Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів” полягає у набутті ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів С/С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

Лабораторна робота виконувалась згідно «Порядок виконання лабораторної роботи та методичні вказівки». Лабораторна робота № 9 стала другою лабораторною роботою з використанням системи контроля версій – GitHub. Для вирішення всіх завдань була взята статична бібліотека з лабораторної роботи № 8 libModulesPoliakh.a

Перше завдання полягало у балі хвилювання моря. Дана умова задачі: висота хвилю в метрах і потрібно було на виході бал хвилювання моря.За допомогою оператора умови if, було створено функцію для рахунку бала, яка знаходиться в ModulesPoliakh.

Друге завдання полягало у створенні функції конвертування температури.Дана умова задачі:температура за шкалою Фаренгейта і потрібно було конвертувати за шкалою Цельсію. Знаючи температуру за шкалою Фаренгейта,температура за шкалою Цельсія розраховувалась наступним чином: tc = 5 / 9 (tf  - 32).Складнощів в завданні не виявилось, адже подібне завдання було в лабораторної роботи №6 з дисципліни «Базові методології та технології програмування І семестр», де потрібно було конвертувати з одної шкали в іншу використовуючи формулу.

Третє завдання полягало у створенні функції для виведення кількості двійкових нулів, або одиниць в складності від результату порівняння. Для задачі потрібно підключення директиви bitset. Директива для мене несподіванкою, адже ніколи не користувався нею,інформацію для вивчення директиви використував інтернет-джерела.

До статичної бібліотеки створеної в лабораторній роботі №8, додав три нові функції, що реалізують розв’язання задачі 9.1, 9.2 та 9.3. Також до заголовкового файлу включив прототипи функцій.

Створив консольний застосунок TestDriver, в якому реалізував модульне тестування функцій за допомогою розроблених наборів контрольних прикладів. Підключив заголовковий файл ModulesPoliakh.h для використання функцій, що знаходяться в статичній бібліотеці. З підключення проблем не виникло,адже воно використовувалось в лабораторній роботі №8.

Для перевірки результату використовув функції завдань.Спочатку створив масив:результат і контрольні приклади. Що б почергово передавати значення у функцію, використав арифметичний цикл for, так як знав скільки ітерацій потрібно програмі пройти. В ході вирішення цієї задачі було виконано аналіз умови та вимог до програмного забезпечення. При виконанні цього завдання використовувалися вже створенні модулі.До всіх завдань було створено TestSuite 9.1-9.3.

Далі потрібно було створити Poliakh\_task в теці prj. Розробити TestSuite до task.Завдання полягало у створенні символів на виклик функції;якщо користувач вводить символ “j” викликається функція s\_calculation (), після чого потрібно ввести x та z. Якщо користувач вводить символ “ z” – викликається функція задачі 9.1, далі вводять кількість висоту хвилювання моря. Якщо вводять символ “с” – функція задачі 9.2, вводить дані про температуру. “j” – функція задачі 9.3, вводимо натуральне число;якщо користувач вводить інші символи,вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “v”, “V” або “A”, відбувається вихід з програми, інакше - виконання програми повторюється. Завдання виявилось не дуже складним,потрібно було трошки пошукати інформацію про звуковий сигнал.

Отже, під час виконання лабораторної роботи № 9 з дисципліни “Базові методології та технології програмування на тему “Реалізація статичних бібліотек модулів лінійних обчислювальних процесів” я набув ґрунтовних вмінь і практичних навичок у використанні статичних, їх програмної реалізації мовою програмування мовою програмування С++ (ISO/IEC 14882:2014) задля реалізації програмних засобів у вільному кросплатформовому Code::Blocks IDE.

**ДОДАТОК А**

**Лістинг програмного забезпечення**

**Лістинг ModulesGordenko 9.1**

#include <iostream>

#include <math.h>

#include <bitset>

using namespace std;

double s\_calculation(double x, double y,double z)

{

return (z + 2 \* pow(y, 2)) / (2.72 \* (x - y)) + sqrt((3.14 \* x));

}

double bal\_baphort(double speed)

{

if (speed < 0.3)

{

return 0;

}

if (speed <= 1.5)

{

return 1;

}

if (speed <= 3.4)

{

return 2;

}

if (speed <= 5.4)

{

return 3;

}

if (speed <= 7.9)

{

return 4;

}

if (speed <= 10.7)

{

return 5;

}

if (speed <= 13.8)

{

return 6;

}

if (speed <= 17.1)

{

return 7;

}

if (speed <= 20.7)

{

return 8;

}

if (speed <= 24.4)

{

return 9;

}

if (speed <= 28.4)

{

return 10;

}

if (speed <= 32.6)

{

return 11;

}

if (speed >= 32.7)

{

return 12;

}

return 0;

}

double temperature(double celciya)

{

return celciya \* 1.8 + 32;

}

int bits\_number(int number)

{

bitset<32> b\_number{number};

if(b\_number[14] == 1){

return b\_number.count();

}

return 32 - b\_number.count();

}

**Gordenko\_Task**

#include <iostream>

#include <locale.h>

#include "ModulesGordenko.h"

using namespace std;

void s\_cal()

{

int x = 0;

int y = 0;

int z = 0;

cout << "Enter X: ";

cin >> x;

cout << "Enter Y: ";

cin >> y;

cout << "Enter Z: ";

cin >> z;

cout << "Result: " << s\_calculation(x, y, z) << endl;

}

void temp()

{

double celcia = 0;

cout << "Enter temperature: ";

cin >> celcia;

cout << endl << "Celsia: " << celcia << endl;

cout << "Farengeit: " << temperature(celcia) << endl;

}

void bal()

{

double speed = 0;

cout << "Enter speed wind: ";

cin >> speed;

cout << "Result: " << bal\_baphort(speed);

}

void bit()

{

int num = 0;

cout << "Enter number: ";

cin >> num;

cout << "Result: " << bits\_number(num) << endl;

}

int main()

{

setlocale (LC\_ALL, "UKR");

int symbol = 0;

char sym;

while(true)

{

cout << "Enter number: ";

cin >> symbol;

if (symbol == 7)

{

s\_cal();

}

if (symbol == 5)

{

bal();

}

if (symbol == 4)

{

temp();

}

if (symbol == 3)

{

bit();

}

else

{

cout << "\a";

}

cout << endl;

cout << "Action: ";

cin >> sym;

if (sym == 'T' || sym == 't' || sym == 'C')

break;

system("pause");

system("cls");

}

}

**ДОДАТОК В**

**(TestSuite 9.1-9.3)**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | US\_9\_1 |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Андрій Горденко |
| Виконавець  Implementer | Андрій Горденко |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TS\_1 | wHeight = 35 | Result = 12 | Passed |
| TS\_2 | WHeight = 7 | Result = 4 | Passed |
| TS\_3 | WHeight = 1.9 | Result = 2 | Passed |
| TS\_4 | WHeight = 0.6 | Result = 1 | Passed |
| TS\_5 | WHeight = 9.6 | Result = 5 | Passed |

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | US\_9\_2 |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Андрій Горденко |
| Виконавець  Implementer | Андрій Горденко |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TS\_1 | Сelsia = 20 | Result = 68 | Passed |
| TS\_2 | Celsia = 5 | Result = 41 | Passed |
| TS\_3 | Celsia = 10 | Result = 50 | Passed |

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | US\_9\_3 |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Андрій Горденко |
| Виконавець  Implementer | Андрій Горденко |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TS\_1 | Number = 10 | Result = 30 | Passed |
| TS\_2 | Number = 30 | Result = 28 | Passed |
| TS\_3 | Number = 25434 | Result = 8 | Passed |
| TS\_4 | Number = 64131 | Result = 9 | Passed |

**Додаток Г**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_4 |
| Рівень тестування  Level of Testing | системне / System Testing |
| Автор тест-сьюта  Test Suite Author | Андрій Горденко |
| Виконавець  Implementer | Андрій Горденко |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування /  Test Result |
| TS\_1 | * Запустити застосунок * Введіть символ:7 * Ввести:6 * Ввести:5 * Ввести:9 | 1.Result:26  2.Для продовження натисніть любую клавишу. | Passed |
| TS\_2 | * Запустити застосунок * Введіть символ:5 * Ввести:7 | 1.Result:14  2.Для продовження натисніть любую клавішу | Passed |
| TS\_3 | * Запустити застосунок * Введіть символ:5 * Ввести: 20 | 1.Farengeit:68  2.Для продовження натисніть любую клавишу. | Passed |
| TS\_4 | * Запустити застосунок * Введіть символ:3 * Ввести:30 | 1.Result:28  2.Для продовження натисніть любую клавішу | Passed |
| TS\_5 | 1.Запустити застосунок  2.Введіть символ:C | 1.Exit | Passed |