Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 3

з навчальної дисципліни “Модульне програмування”

реалізація програмних модулів розгалужених  
та ітераційних обчислювальних процесів.

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КБ-19

Москальов А.В.

ПЕРЕВІРИВ

старший викладач кафедри кібербезпеки та програмного забезпечення

Поліщук Л. І.

Кропивницький – 2020

Тема: реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів..

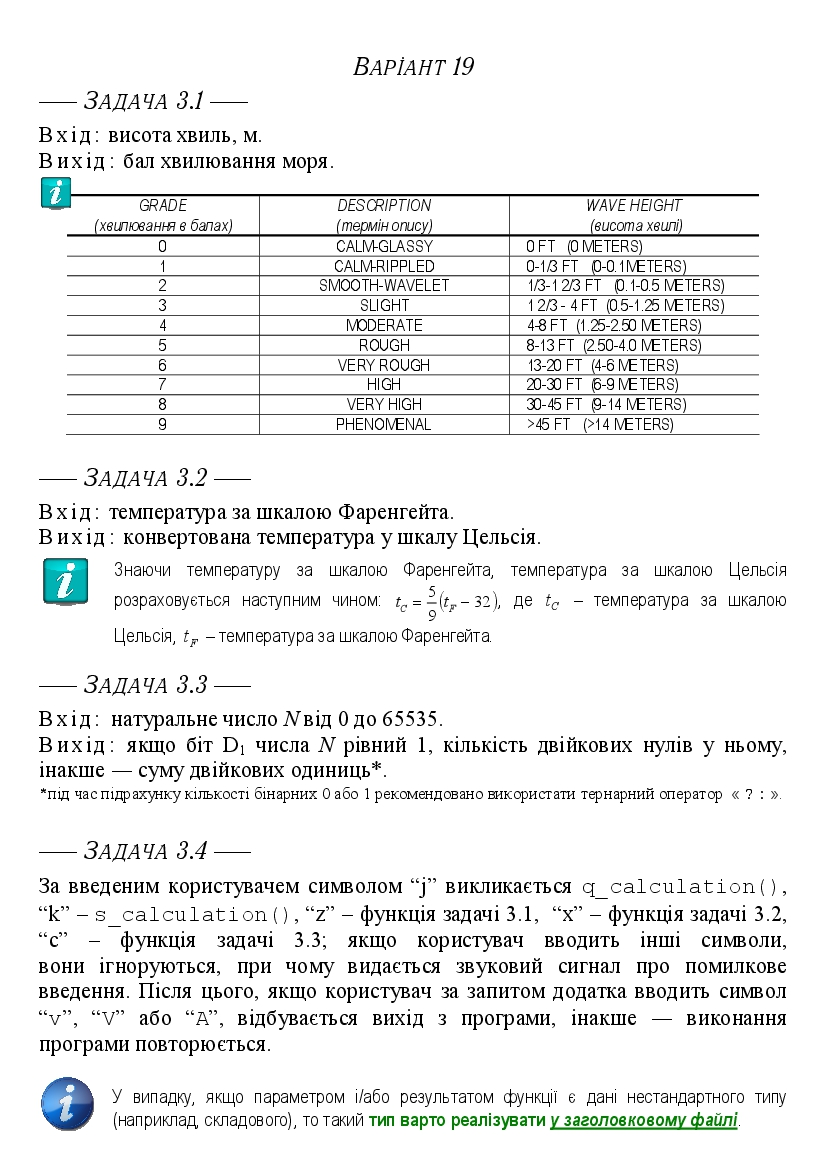
Мета: набуття ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів С/С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому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

Завдання:

1. Реалізувати функції розв’язування задач 3.1–3.3 як складових статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище, створений під час виконання лабораторної роботи №2).

2. Реалізувати програмне забезпечення розв’язування задачі 3.4 на основі функцій статичної бібліотеки libModulesПрізвище.а.

Варіант №19:



**Аналіз задачі 3.1:**

Вхідні дані рядкового типу (задля валідації). Значення не може бути меншим за 0. Результат перевірятиметься кейсом з відповідністю.

**Аналіз задачі 3.2:**

Вхідні дані рядкового типу (задля валідації). Значення може бути в діапазоні від -22 до +122. Вхідний результат буде дійсним, вихід також. До потоку виводу буде включено температуру у цельсіях.

**Аналіз задачі 3.3:**

Вхідні дані рядкового типу (задля валідації). В подальшому число буде натуральним, в діапазоні від 0 до 65535. Після обробки вивід буде залежить від біту D1. В разі, якщо цей біт == 1, то буде кількість двійкових нулів у числі, інакше – суму двійкових одиниць.

**Результат виконання тестового драйвера до задач 3.1 – 3.3:**

|  |  |  |
| --- | --- | --- |
| Задача 3.1 | Задача 3.2 | Задача 3.3 |
| 3.1 testing...  Test #0  Input:0  Result:0  Status: Passed.  Test #1  Input:0.05  Result:1  Status: Passed.  Test #2  Input:0.4  Result:2  Status: Passed.  Test #3  Input:1  Result:3  Status: Passed.  Test #4  Input:1.25  Result:3  Status: Passed.  Test #5  Input:2  Result:4  Status: Passed.  Test #6  Input:3.5  Result:5  Status: Passed.  Test #7  Input:5  Result:6  Status: Passed.  Test #8  Input:7  Result:7  Status: Passed.  Test #9  Input:47  Result:9  Status: Passed. | 3.2 testing...  Test #0  Input:32  Result:0  Status: Passed.  Test #1  Input:50  Result:10  Status: Passed.  Test #2  Input:77  Result:25  Status: Passed.  Test #3  Input:-22  Result:-30  Status: Passed.  Test #4  Input:122  Result:50  Status: Passed.  Test #5  Input:14  Result:-10  Status: Passed.  Test #6  Input:5  Result:-15  Status: Passed.  Test #7  Input:41  Result:5  Status: Passed.  Test #8  Input:59  Result:15  Status: Passed.  Test #9  Input:68  Result:20  Status: Passed. | 3.3 testing...  Test #0  Input:11  Result:O2  Status: Passed.  Test #1  Input:433  Result:O5  Status: Passed.  Test #2  Input:332  Result:N5  Status: Passed.  Test #3  Input:3222  Result:O6  Status: Passed.  Test #4  Input:122  Result:O5  Status: Passed.  Test #5  Input:14  Result:O3  Status: Passed.  Test #6  Input:5  Result:O2  Status: Passed.  Test #7  Input:41  Result:N3  Status: Passed.  Test #8  Input:59  Result:O5  Status: Passed.  Test #9  Input:68  Result:N5  Status: Passed. |

**Вихідний код проекта TestDriver.cpp та ModulesMoskalev.cpp:**



**Аналіз задачі 3.4:**

Основні функції будуть виконувати вже готові модулі. Виключенням є лише меню з вибором та вихід з ПЗ. Для цього використовую string, щоб перевіряти дані, навіть якщо введено більше одного символа.

**Код задачі 3.4 (Moskalev\_task):**

#include <iostream>

#include <string>

#include <iomanip>

#include <sstream>

#include <ModulesMoskalev.h>

#include <stdio.h>

using namespace std;

string ValidCh(string X) {

bool znak = false;

string chisla = "0123456789,.-";

for (short i=0; i<X.length(); i++){

if (chisla.find(X[i]) == -1) {return "N";}

}

for (short i=0; i<X.length(); i++){

if (X[i] == ',') {X[i] = '.';}

}

for (short i=0; i<X.length(); i++){

if (X[i] == '.' && znak == true) {return "N";}

if (X[i] == '.' && znak == false) {znak = true;}

}

if (X[0] == '.' || X[X.length()-1] == '.') {return "N";}

for (short i=1; i<X.length(); i++){

if (X[i] == '-') {return "N";}

}

return X;

}

void zad34j(){

string xs,ys,zs;

float x,y,z;

Start\_Input\_X:

do{

cout << endl << "Задайте X: ";

cin >> xs;

if (ValidCh(xs) == "N") {cout << "Некоректне число!";}

} while (ValidCh(xs) == "N");

istringstream(ValidCh(xs)) >> x;

if (x<1 || x>100) {cout << "Допустимий дiапазон: 1-100!"; goto Start\_Input\_X;}

Start\_Input\_Y:

do{

cout << endl << "Задайте Y: ";

cin >> ys;

if (ValidCh(ys) == "N") {cout << "Некоректне число!";}

} while (ValidCh(ys) == "N");

istringstream(ValidCh(ys)) >> y;

if (y<1 || y>100) {cout << "Допустимий дiапазон: 1-100!"; goto Start\_Input\_Y;}

Start\_Input\_Z:

do{

cout << endl << "Задайте Z: ";

cin >> zs;

if (ValidCh(zs) == "N") {cout << "Некоректне число!";}

} while (ValidCh(zs) == "N");

istringstream(ValidCh(zs)) >> z;

if (z<1 || z>100) {cout << "Допустимий дiапазон: 1-100!"; goto Start\_Input\_Z;}

cout << "q\_caluclation: " << q\_calculation(x,y,z) << endl;

}

void zad34k(){

string xs,ys,zs;

float x,y,z;

Start\_Input\_X:

do{

cout << endl << "Задайте X: ";

cin >> xs;

if (ValidCh(xs) == "N") {cout << "Некоректне число!";}

} while (ValidCh(xs) == "N");

istringstream(ValidCh(xs)) >> x;

if (x<1 || x>100) {cout << "Допустимий дiапазон: 1-100!"; goto Start\_Input\_X;}

Start\_Input\_Y:

do{

cout << endl << "Задайте Y: ";

cin >> ys;

if (ValidCh(ys) == "N") {cout << "Некоректне число!";}

} while (ValidCh(ys) == "N");

istringstream(ValidCh(ys)) >> y;

if (y<1 || y>100) {cout << "Допустимий дiапазон: 1-100!"; goto Start\_Input\_Y;}

Start\_Input\_Z:

do{

cout << endl << "Задайте Z: ";

cin >> zs;

if (ValidCh(zs) == "N") {cout << "Некоректне число!";}

} while (ValidCh(zs) == "N");

istringstream(ValidCh(zs)) >> z;

if (z<1 || z>100) {cout << "Допустимий дiапазон: 1-100!"; goto Start\_Input\_Z;}

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

}

void zad34z(){

string xs;

int x;

Start\_Input\_X:

do{

cout << '\n' <<"Задайте висоту хвилi(М):";

cin >> xs;

if (ValidCh(xs) == "N") {cout << "Некоректне число!";}

} while (ValidCh(xs) == "N");

istringstream(ValidCh(xs)) >> x;

if (x<0) {cout << "Значення не може бути менше нуля."; goto Start\_Input\_X;}

cout << "Бал хвилювання моря:" << WaterHeight(x) << endl;

}

void zad34x(){

string xs;

int x;

Start\_Input\_X:

do{

cout << '\n' <<"Задайте температуру в форенгейтах:";

cin >> xs;

if (ValidCh(xs) == "N") {cout << "Некоректне число!";}

} while (ValidCh(xs) == "N");

istringstream(ValidCh(xs)) >> x;

if (x<-22 || x>122) {cout << "Допустимий дiапазон: -22 .. 122."; goto Start\_Input\_X;}

cout << "Температура в цельсiях:" << ForToCels(x) << endl;

}

void zad34c(){

string xs;

int x;

Start\_Input\_X:

do{

cout << '\n' <<"Задайте число:";

cin >> xs;

if (ValidCh(xs) == "N") {cout << "Некоректне число!";}

} while (ValidCh(xs) == "N");

istringstream(ValidCh(xs)) >> x;

if (x<0 || x>65535) {cout << "Допустимий дiапазон: 0-65535"; goto Start\_Input\_X;}

int i;

int c;

if (zadacha33(x)[0] == 'N') {

cout << "D1 = 0, кiлькiсть нулiв: ";

for (i=1; i<zadacha33(x).length(); i++){

cout << zadacha33(x)[i];}

}

if (zadacha33(x)[0] == 'O') {

cout << "D1 = 1, сума одиниць: ";

for (i=1; i<zadacha33(x).length(); i++){

cout << zadacha33(x)[i];}

}

cout << '\n';

}

int main(){

setlocale(LC\_ALL, "Ukrainian");

cout << " ----=== Лабораторна робота 3. ===----"<< '\n';

MenuV:

cout << "Оберiть бажану дiю: " << endl;

cout << "j - q\_calculation." << '\n' << "k - s\_calculation." << '\n' << "z - Задача 3.1." << '\n' << "x - Задача 3.2." << '\n' << "с - Задача 3.3." << '\n' <<"Вiдповiдь:";

string menu1;

cin >> menu1;

if (menu1.length() != 1) {cout << "Оберiть коректний варiант!"; goto MenuV;}

cout << '\n';

switch (menu1[0]) {

case 'j':

zad34j(); break;

case 'k':

zad34k(); break;

case 'z':

zad34z(); break;

case 'x':

zad34x(); break;

case 'c':

zad34c(); break;

default:

cout << '\a'; goto MenuV; break;

}

MenuE:

cout << "Бажаєте повторити запит?" << '\n' << "v, V, A - вихiд. Iнакши - повтор: ";

cin >> menu1;

if (menu1.length() != 1) {cout << "Оберiть коректний варiант!"; goto MenuE;}

cout << '\n';

switch (menu1[0]) {

case 'v':

return 0; break;

case 'V':

return 0; break;

case 'A':

return 0; break;

default:

goto MenuV; break;

}

}

Висновок:

В ході виконання лабораторної роботи я працював з модулями, зробленими раніше. Це дуже зручно та економить багато часу, адже мені не було необхідності переписувати модулі спочатку. Також ці модулі ми доповнювали. Цей процес не викликав ніяких проблем. Я просто відкрив модуль, описав нові функції у .h файлі, а потім розробив ці функції у .cpp файлі. Самі завдання були досить простими, окрім задачі 3.3. В задачі 3.1 за допомогою if`а я перевіряв значення числа та вивод відповідну інформацію. Слід також зауважити, що вказуючи діапазони x<3 && x>5 та x<5 && x>7 (наприклад) слід не забути, що число може бути цілим, наприклад x==5, тому в першому випадку слід x>5 замінити на x>=5 задля запобігання проблем. Задача 3.2 теж була легкою в реалізації. Просто використавши формулу та поставивши інтервали (можна було використати функції з 2ї лабораторної роботи) ми отримували необхідний результат, що і виводили. В задачі 3.3 виникли деякі труднощі. Для переводу числа з десяткового до двійкового мною було використано формулу, що я дізнався на уроках інформатики: постійне ділення на 2 та запис остачі. Після цього результат було записано у рядок, який аналізувався на значення D1, у випадку, якщо число виводило 0 – за допомогою циклу підраховувалась кількість нулів у двійковому коді. Якщо ж D1==1, то всі одиниці двійкового коду приплюсовувались, також за допомогою циклу. Функція, задля спрощення подальшого використання, виводило першим символом O або N, що означало значення D1, одиниця або нуль відповідно, а потім кількість нулів або суму одиниць. Потім, маючи готові функції, було створено консольну програму, що використовувала модуль, написаний раніше та мала в собі інтерфейс, яким міг керувати користувач. Користувач міг обрати, яку задачу в поточний момент почати вирішувати, а також, в разу некоректного вводу, виводилось звукове попередження. Після вибору необхідної задачі користувача переводило на модуль, що відповідав обраній задачі, де він продовжував з нею працювати (вводити значення, отримувати результат). По завершенню роботи з модулем користувачеві пропонувалось або повторити виконання, або завершити виконання програмного забезпечення. Обидва ці меню були виконані за допомогою оператору switch.