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”

На тему:

“ Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів”

ВИКОНАВ

студент академічної групи КБ-21

\_\_\_\_\_\_\_\_\_ К.О. Задорожний

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_ П.С. Усік

м. Кропивницький 2022

**Мета роботи :**

полягає у набутті ґрунтовних вмінь і практичнихнавичок реалізації технології модульного програмування,застосування операторів С/С++ арифметичних, логічних, побітовихоперацій, умови, циклів та вибору під час розроблення статичнихбібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

**Завдання :**

1. Реалізувати функції розв’язування задач 9.1–9.3 як складовихстатичної 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 :** №14

**ЗАДАЧА 9.1**

Відповідно до трудового договору, працівникові нараховується надбавка до заробітної плати за стаж роботи на підприємстві наступним чином: за 2-5 років стажу надбавка складає 5%; за 5-10 років — 10%; понад 10 років – 25%.

**Вхід:** розмір заробітної плати працівника (грн.) та стаж його роботи на підприємстві (років).

**Вихід:** розмір надбавки (%), сума до виплати (грн., включно з надбавкою).

**Аналіз умови до задачі 9.1**

Користувач вводить свою заробітну плату в грн. та стаж роботи. Після цього потрібно вивести на екран розмір його надбавки та суму до виплати включно з надбавкою(відсоткова ставка) в залежності від стажу його роботи. Заробітна плата і стаж роботи повинні буди додатними числами.

**Алгоритм до задачі 9.1**

**START**

1. PRINT “Enter your salary”;
2. INPUT salary;
3. PRINT “Enter your work experience”;
4. INPUT workExp;
5. IF 2 ≤ workExp < 5 THEN

PRINT “ “allowance” - 5 %, “salary” – salary\*1,05”

1. ELSE IF 5 ≤ workExp < 10

PRINT “ “allowance” - 10 %, “salary” – salary\*1,10”

1. ELSE IF 10 ≤ workExp

PRINT “ “allowance” - 25 %, “salary” – salary\*1,25”

1. ELSE

PRINT “ “allowance” – 0 %, “salary” – salary”

1. ENDIF

**END**

**ЗАДАЧА 9.2**

**Вхід:** довжина обхвату голови у сантиметрах (числа від 53 до 62 включно).

**Вихід:** розмір шолому.

Таблиця 1 - **Розміри шоломів**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| РОЗМІР | XS | S | M | L | XL |
| ДОРОСЛІ, см | 53-54 | 55-56 | 57-58 | 59-60 | 61-62 |
| ДІТИ, см | - | 48-49 | 50-51 | 52-53 | - |

**Аналіз задачі 9.2**

Користувач вводить розмір обхвату голови у сантиметрах, від 53 до 62 включно. І вивести розмір шолому відповідно таблиці 1

**Алгоритм до задачі 9.2**

**START**

1. PRINT “Enter your head circumference in centimeters”;
2. INPUT size;
3. IF 53 ≤ size ≤ 54 THEN

PRINT “Helmet size - XS”

1. ELSE IF 55 ≤ size ≤ 56

PRINT “Helmet size - S”

1. ELSE IF 57 ≤ size ≤ 5

PRINT “Helmet size - M”

1. ELSE IF 59 ≤ size ≤ 60

PRINT “Hemet size - L”

1. ELSE IF 61 ≤ size ≤ 61

PRINT “Helmet size - XL”

1. ELSE

PRINT “Helmet size – not found”

1. ENDIF

**END**

**ЗАДАЧА 9.3**

**Вхід**: натуральне число N від 0 до 40008050.

**Вихід**: якщо біт D15 числа N рівний 1, кількість двійкових нулів у ньому, інакше - кількість двійкових одиниць\*.

\*під час підрахунку кількості бінарних 0 або 1 рекомендовано використати тернарний оператор «? :».

**Аналіз задачі 9.3**

Користувач вводить число від 0 до 40008050. Спочатку число потрібно перетворити в двійкову систему числення. Потім після цього порівняти 15 біт, якщо він дорівнює 0, то вивести кількість двійкових одиниць, інакше – кількість двійкових нулів.

**Алгоритм до задачі 9.2**

**START**

1. SET number
2. SET countOne
3. SET binaryNumber[32]
4. PRINT “Enter number”;
5. INPUT number;
6. PRECONDITION: зміна і дорівнює 0;
7. FOR поки number > 0

binaryNumber[i] := number %2;

IF number%2 == 1 THEN;

countOne := countOne + 1;

ENDIF

number := number/2;

i := i + 1;

1. ENDWHILE
2. IF binaryNumber[15] = 1 THEN

PRINT "number of binary ones " countOne;

1. ELSE

PRINT "number of binary zeros " 32 - countOne

1. ENDIF

**END**

**Результати виконання тестового модуля**

Test task 9.1

Test is passed

Test is passed

Test is passed

Test is passed

Test is passed

Test task 9.2

Test is passed

Test is passed

Test is passed

Test is passed

Test is passed

Test task 9.3

Test is passed

Test is passed

Test is passed

Test is passed

Test is passed

**ЗАДАЧА 9.4**

За введеним користувачем символом “е” викликається s\_calculation (), “r” – функція задачі 9.1, “f” – функція задачі 9.2, “t” – функція задачі 9.3; якщо користувач вводить інші символи, вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “g”, “G” або “о”, відбувається вихід з програми, інакше виконання програми повторюється.

**Аналіз задачі 9.4**

Користувач вводить символ с клавіатури, якщо цей символ “е” викликається s\_calculation (), “r” – функція задачі 9.1, “f” – функція задачі 9.2, “t” – функція задачі 9.3; якщо користувач вводить інші символи, вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “g”, “G” або “о”, відбувається вихід з програми, інакше виконання програми повторюється.

**Алгоритм до задачі 9.3**

**START**

1. WHILE поки action != ‘g’ and action != ‘G’ and action != ‘o’
2. PRINT “Enter action”;
3. INPUT action;
4. IF action = ‘e’ THEN;

s\_calculate();

1. ELSE IF action = ‘r’

task\_9\_1();

1. ELSE IF action = ‘f’

task\_9\_2();

1. ELSE IF action = ‘t’

task\_9\_3();

1. ELSE

PRINT Звуковий сигнал

1. ENDWHILE

**END**

**Висновок**

Під час виконання лабораторної роботи № 9 на тему **«**Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів**»** з предмету «Базові Методології та Технології Програмування» я відпрацював на практиці створення статичних бібліотек та власних модулів і заголовних файлів, написання власних текстових драйверів для своїх модулів. Набув 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.

Під час лабораторної роботи було виконано такі пункти:

* Завантажено завдання №9 із репозиторію за допомогою команди git pull
* Після цього було створено папки prj, Software, Test Suite, та Report і заповнено README.md файл.
* Після цього було відправлено ці дані за допомогою команд :

git add -A

git commit -m “Add folder prj, Software, Test Suite, Report and feeling README file”

git push

* Прочитано завдання та виконано декомпозицію задачі 9.1
* Виконано аналіз першого завдання, в результаті чого, створено файл–шаблон тестового набору (Unit testing) із назвою UT\_9\_1.
* Розроблено 5 Test Case
* У Code::Blocks створено статичну бібліотеку із назвою ModulesZadorozhny
* За допомогою визначених вимог до задачі та розробленого алгоритму було реалізовано функцію task\_9\_1 () і поміщено прототип функції в заголовний файл ModulesZadorozhny.h.
* Прочитано завдання та виконано декомпозицію задачі 9.2
* Виконано аналіз другого завдання, в результаті чого, створено файл–шаблон тестового набору (Unit testing) із назвою UT\_9\_2.
* Розроблено 5 Test Case
* За допомогою визначених вимог до задачі та розробленого алгоритму було реалізовано функцію task\_9\_2 () і поміщено прототип функції в заголовний файл ModulesZadorozhny.h.
* Прочитано завдання та виконано декомпозицію задачі 9.3
* Виконано аналіз другого завдання, в результаті чого, створено файл–шаблон тестового набору (Unit testing) із назвою UT\_9\_3.
* Розроблено 5 Test Case
* За допомогою визначених вимог до задачі та розробленого алгоритму було реалізовано функцію task\_9\_3 () і поміщено прототип функції в заголовний файл ModulesZadorozhny.h.
* Потім було написаний тестовий драйвер для функції task\_9\_1(), task\_9\_2(), task\_9\_3() та протестовано із всіма позначками passed.
* Після компіляції TestDriver було створено TestDriver.exe . Цей файл був переміщенний в папку Software.
* Після компіляції Zadorozhny\_task було створено Zadorozhny\_task.exe . Цей файл був переміщенний в папку Software.
* Виконано системне тестування ПЗ застосунку Zadorozhny\_task.exe, результати записані у TS\_9\_4, всі тести були пройдені із результатом passed.
* Виконано системне тестування ПЗ застосунку Zadorozhny\_task.exe, результати записані у TS\_8\_2, всі тести були пройдені із результатом passed.
* Всі данні було відправлено на репозиторій в папку lab09

Було цікаво працювати із бітами. В ДОДАТКУ А указані лістинг програм, статичної бібліотеки, тестового драйвера та Zadorozhny\_task. В ДОДАТКУ Б указані знаходяться Unit test до функції task\_9\_1(), task\_9\_2(), task\_9\_3(), та Test Suite до задачі 9.4

**ДОДАТОК А**

**Модульний тест до задачі 9.1**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_9\_1 |
| Назва модуля  Name of Modules/ prj / ModulesZadorozhny | task\_9\_1() |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Задорожний Костянтин |
| Виконавець  Implementer | Задорожний Костянтин |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Значення | Очікуваний результат | Результат модульного тестування  (passed/fail) |
| 1 | Salary = 10000  Work Experience = 3 | 10500 | passed |
| 2 | Salary = 10000  Work Experience = 6 | 11000 | passed |
| 3 | Salary = 10000  Work Experience = 11 | 12500 | passed |
| 4 | Salary = 10000  Work Experience = 1 | 10000 | passed |
| 5 | Salary = 100000  Work Experience = 30 | 125000 | passed |

**Модульний тест до задачі 9.2**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_9\_2 |
| Назва модуля  Name of Modules/ prj / ModulesZadorozhny | task\_9\_2() |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Задорожний Костянтин |
| Виконавець  Implementer | Задорожний Костянтин |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Значення | Очікуваний результат | Результат модульного тестування  (passed/fail) |
| 1 | size\_head = 53 | XS | passed |
| 2 | size\_head = 55 | S | passed |
| 3 | size\_head = 58 | M | passed |
| 4 | size\_head = 60 | L | passed |
| 5 | size\_head = 61 | XL | passed |

**Модульний тест до задачі 9.3**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | UT\_9\_3 |
| Назва модуля  Name of Modules/ prj / ModulesZadorozhny | task\_9\_3() |
| Рівень тестування  Level of Testing | модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Задорожний Костянтин |
| Виконавець  Implementer | Задорожний Костянтин |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Значення | Очікуваний результат | Результат модульного тестування  (passed/fail) |
| 1 | number = 58 | 28 | passed |
| 2 | number = 39854 | 10 | passed |
| 3 | number = 67009 | 26 | passed |
| 4 | number = 40008050 | 19 | passed |
| 5 | number = 3987532 | 11 | passed |

**Системний тест до задачі 9.4**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_4 |
| Назва проекта / ПЗ  Name of Project / Software | Zadorozhny\_task |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Задорожний Костянтин |
| Виконавець  Implementer | Задорожний Костянтин |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| TS-1 | 1. Ввести “e” 2. Ввести 12 3. Ввести 30 4. Ввести 5 5. Ввести “o” | Enter action  ==>e  Enter x  ==>12  Enter y  ==>30  Enter z  ==>5  (3 \* sin \* √(12 \* 12 + lg(12 - 3)))^30 + 5/12 = 13334.549805  Enter action  ==>o | passed |
| TS-2 | 1. Ввести “r” 2. Ввести 20000 3. Ввести 20 4. Ввести “g” | Enter action  ==>r  Enter your salary  ==>20000  Enter your work experience  ==>20  Allowance - 25%, salary - 25000.000000  Enter action  ==>g | passed |
| TS-3 | 1. Ввести “f” 2. Ввести 54 3. Ввести “G” | Enter action  ==>f  Enter your head circumference in centimeters  ==>54  Helmet size - XS  Enter action  ==>G | passed |
| TS-4 | 1. Ввести “t” 2. Ввести 4568 3. Ввести “g” | Enter action  ==>t  Enter number  ==>4568  26  Enter action  ==>g | passed |
| TS-5 | 1. Ввести “q” 2. Ввести “l” 3. Ввести “o” | Enter action  ==>q  \*Звуковий сигнал\*  Enter action  ==>l  \*Звуковий сигнал\*  Enter action  ==>o | passed |

**ДОДАТОК Б**

**Лістенінг програми ModulesZadorozhny**

#include <iostream>

#include <math.h>

#include <tgmath.h>

#include <windows.h>

#include <bitset>

#include <string>

using namespace std;

float s\_calculation(int x, int y, int z)

{

return (pow(3 \* sin(sqrt(12 \* x + log(x - 3))), y) + z/x);

}

string s\_calculation\_to\_string(int x, int y, int z){

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

return "(3 \* sin \* √(12 \* " + to\_string(x) + " + lg(" + to\_string(x) + " - 3)))^" +

to\_string(y) + " + " + to\_string(z) + "/" + to\_string(x) + " = " + to\_string(s\_calculation(x, y, z));

}

int task\_9\_1(int salary, int work\_exp){

if(work\_exp >= 2 && work\_exp < 5){

return salary \* 1.05;

}

else if(work\_exp >= 5 && work\_exp < 10){

return salary \* 1.1;

}

else if(work\_exp >= 10){

return salary \* 1.25;

}

else{

return salary;

}

}

string head\_task\_9\_1(int salary, int work\_exp){

if(work\_exp >= 2 && work\_exp < 5){

return "Allowance - 5%, salary - " + to\_string(salary \* 1.05);

}

else if(work\_exp >= 5 && work\_exp < 10){

return "Allowance - 10%, salary - " + to\_string(salary \* 1.1);

}

else if(work\_exp >= 10){

return "Allowance - 25%, salary - " + to\_string(salary \* 1.25);

}

else{

return "Allowance - 0%, salary - " + to\_string(salary);

}

}

string task\_9\_2(int size\_head){

switch(size\_head){

case 53:

case 54:

return "XS";

case 55:

case 56:

return "S";

case 57:

case 58:

return "M";

case 59:

case 60:

return "L";

case 61:

case 62:

return "XL";

default :

return "not found";

}

}

int task\_9\_3(int number){

bitset<32> b\_number(number);

if(b\_number[15]){

return b\_number.count();

}

else {

return 32 - b\_number.count();

}

}

string test\_module(bool result){

if(result){

return "Test is passed";

}

else{

return "Test failed";

}

}

**Лістинг TestDriver**

#include "../ModulesZadorozhny.h"

int main(){

int salary[5]= {10000, 10000, 10000, 10000, 100000};

int work\_exp[5] = {3, 6, 11, 1, 30};

int result\_task\_9\_1[5] = {10500, 11000, 12500, 10000 , 125000};

int size\_head[5] = {53, 55, 58, 60, 61};

string result\_task\_9\_2 [5] = {"XS", "S", "M", "L", "XL"};

int number[5] = {58, 39854, 67009, 40008050, 3987532};

int result\_task\_9\_3[5] = {28, 10, 26, 19, 11};

cout << "Test task 9.1" << endl;

for(int i = 0; i < 5; i++){

cout << test\_module( result\_task\_9\_1[i] == task\_9\_1(salary[i], work\_exp[i])) << endl;

}

cout << "Test task 9.2" << endl;

for(int i = 0; i < 5; i++){

cout << test\_module(result\_task\_9\_2[i] == task\_9\_2(size\_head[i])) << endl;

}

cout << "Test task 9.3" << endl;

for(int i = 0; i < 5; i++){

cout << test\_module(result\_task\_9\_3[i] == task\_9\_3(number[i])) << endl;

}

return 0;

}

**Лістинг Zadorozhny\_task**

#include "../ModulesZadorozhny.h"

int main()

{

int x = 0.0;

int y = 0.0;

int z = 0.0;

int salary = 0;

int work\_exp = 0;

int size\_head = 0;

int number = 0;

char action;

do{

cout << "Enter action" << endl;

cout << "==>";

cin >> action;

if(action == 'e'){

cout << "Enter x" << endl;

cout << "==>";

cin >> x;

cout << "Enter y" << endl;

cout << "==>";

cin >> y;

cout << "Enter z" << endl;

cout << "==>";

cin >> z;

cout << s\_calculation\_to\_string(x, y, z) << endl;

}

else if(action == 'r'){

cout << "Enter your salary" << endl;

cout << "==>";

cin >> salary;

cout << "Enter your work experience" << endl;

cout << "==>";

cin >> work\_exp;

cout << head\_task\_9\_1(salary, work\_exp) << endl;

}

else if(action == 'f'){

cout << "Enter your head circumference in centimeters" << endl;

cout << "==>";

cin >> size\_head;

cout << "Helmet size - ";

cout << task\_9\_2(size\_head) << endl;

}

else if(action == 't'){

cout << "Enter number" << endl;

cout << "==>";

cin >> number;

cout << task\_9\_3(number) << endl;

}

else if(action != 'g' && action != 'G' && action != 'o'){

cout << '\a' << endl;

}

}while(action != 'g' && action != 'G' && action != 'o');

return 0;

}