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.

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

ВИКОНАВ

студент академічної групи КН-24

Терещенко В.О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А.С.

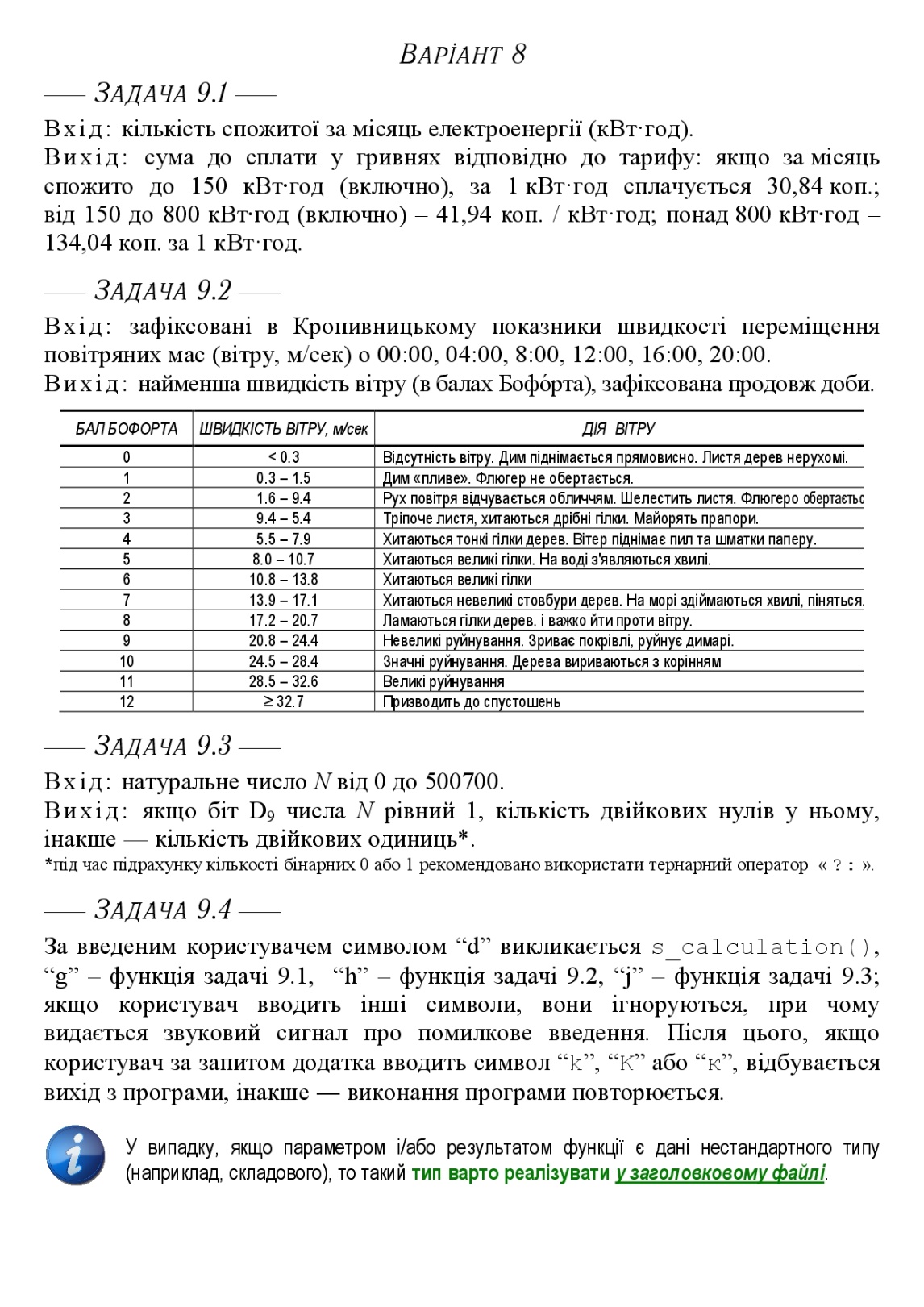
Кропивницький – 2025

Тема:реалізація програмних модулів розгалужених та ітераційних обчислювальних 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 до лаборатоної роботи: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.



**Завдання 9.1**

* **Вхід:** Кількість спожитої енергії за місяць (кВт·год).
* **Вихід:** Сума до сплати в гривнях:
  + До 150 кВт·год (включно): 30,84 коп/кВт·год.
  + Від 150 до 800 кВт·год (включно): 41,94 коп/кВт·год.
  + Понад 800 кВт·год: 134,04 коп за 1 кВт·год.

**Завдання 9.2**

* **Вхід:** Показники швидкості переміщення повітря (м/сек) об 00:00, 04:00, 08:00, 12:00, 16:00, 20:00.
* **Вихід:** Найменша швидкість вітру (в балах Бофорта), зафіксована протягом доби.

**Бал Бофорта:**

| **Швидкість вітру, м/сек** | **Оцінка** |
| --- | --- |
| < 0.3 | Відсутність вітру. |
| 0.3 - 1.5 | Дим піднімається прямо. |
| 1.6 - 4.9 | Рух повітря визначається освітленням. |
| 9.4 - 13.8 | Тривале тихе. |
| 5.5 - 7.9 | Характеристики тонких гілок дерев. |
| 7.9 - 10.7 | Характеристики великих гілок. |
| 13.9 - 17.9 | Характеристики негрибних дерев. |
| 17.9 - 20.7 | Ламкість великих дерев. |
| 20.7 - 24.4 | Значні руйнування. |
| 24.5 - 28.5 | Дерево рідко тримається. |
| 28.5 - 32.7 | Призводити до спустошень. |

**Завдання 9.3**

* **Вхід:** Натуральне число N від 0 до 500700.
* **Вихід:** Якщо біт D9 числа N рівний 1, кількість двійкових нулів у ньому, інакше — кількість дійсових одиниць.

**Завдання 9.4**

* Викликається з символом “d” функція **s\_calculation()**.
* Інші символи ведуть до ігнорування.
* Якщо використати символи “k”, “k?”, відбувається повторюваний виклик програми.

**Лістинг 9.1**

#include <iostream>

using namespace std;

int main() {

double kilowats;

double totalCost = 0.0;

cout << "Enter the number of kilowatt-hours consumed per month: ";

cin >> kilowats;

if (kilowats <= 150) {

totalCost = kilowats \* 0.3084;

} else if (kilowats <= 800) {

totalCost = 150 \* 0.3084 + (kilowats - 150) \* 0.4194;

} else {

totalCost = 150 \* 0.3084 + (800 - 150) \* 0.4194 + (kilowats - 800) \* 1.3404;

}

cout << "Amount due: " << totalCost << " grn" << endl;

return 0;

}

**Лістинг 9.2**

#include <iostream>

using namespace std;

int getBeaufortScale(double wind) {

if (wind < 0.3) return 0;

if (wind < 1.6) return 1;

if (wind < 3.4) return 2;

if (wind < 5.5) return 3;

if (wind < 8.0) return 4;

if (wind < 10.8) return 5;

if (wind < 13.9) return 6;

if (wind < 17.2) return 7;

if (wind < 20.8) return 8;

if (wind < 24.5) return 9;

if (wind < 28.5) return 10;

if (wind < 32.7) return 11;

return 12;

}

int main() {

double wind;

int minBeaufort = 12;

cout << "Введіть значення сили вітру (м/с) у 6 різних годинах:" << endl;

for (int i = 0; i < 6; i++) {

cin >> wind;

int beaufort = getBeaufortScale(wind);

if (beaufort < minBeaufort) {

minBeaufort = beaufort;

}

}

cout << "Найменший бал за шкалою Бофорта протягом доби: " << minBeaufort << endl;

return 0;

}

**Лістинг 9.3**

#include <iostream>

#include <bitset>

int task\_9\_3(int N) {

if ((N >> 9) & 1) {

std::bitset<32> bits(N);

return 32 - bits.count();

} else {

std::bitset<32> bits(N);

return bits.count();

}

}

**TestDriver**

#include <iostream>

#include <string>

#include <vector>

#include <iomanip>

using namespace std;

double calculateElectricityBill(double kilowatts);

int getBeaufortScale(double wind);

int getMinBeaufortScale(const double windSpeeds[], int size);

int analyzeBinary(int N);

void s\_calculation();

int main() {

cout << fixed << setprecision(2);

cout << "Тест 1: Розрахунок рахунку за електроенергію:\n";

double kW[] = {100, 200, 1000};

for (double val : kW) {

cout << " " << val << " кВт: " << calculateElectricityBill(val) << " грн\n";

}

cout << "\nТест 2: Шкала Бофорта для одиничних значень вітру:\n";

double windSpeedsTest[] = {0.1, 2.0, 5.0, 12.0, 25.0, 40.0};

for (double wind : windSpeedsTest) {

cout << " " << wind << " м/с: ступінь " << getBeaufortScale(wind) << "\n";

}

cout << "\nТест 3: Мінімальна шкала Бофорта в масиві:\n";

double windArr[] = {3.5, 6.2, 1.4, 10.0, 2.5};

int minScale = getMinBeaufortScale(windArr, 5);

cout << " Мінімальна шкала Бофорта: " << minScale << "\n";

cout << "\nТест 4: Аналіз бітів у числі:\n";

vector<int> testNums = {256, 511, 512, 1023, 1024};

for (int num : testNums) {

cout << " Число: " << num << ", результат: " << analyzeBinary(num) << "\n";

}

cout << "\nТест 5: Виклик функції s\_calculation():\n";

s\_calculation();

return 0;

}

**Лістинг програми Tereshchenko\_task**

#include <iostream>

#include <bitset>

#include <string>

#include <cctype>

using namespace std;

double calculateElectricityBill(double kilowatts) {

if (kilowatts <= 150) {

return kilowatts \* 0.3084;

} else if (kilowatts <= 800) {

return 150 \* 0.3084 + (kilowatts - 150) \* 0.4194;

} else {

return 150 \* 0.3084 + (800 - 150) \* 0.4194 + (kilowatts - 800) \* 1.3404;

}

}

int getBeaufortScale(double wind) {

if (wind < 0.3) return 0;

if (wind < 1.6) return 1;

if (wind < 3.4) return 2;

if (wind < 5.5) return 3;

if (wind < 8.0) return 4;

if (wind < 10.8) return 5;

if (wind < 13.9) return 6;

if (wind < 17.2) return 7;

if (wind < 20.8) return 8;

if (wind < 24.5) return 9;

if (wind < 28.5) return 10;

if (wind < 32.7) return 11;

return 12;

}

int getMinBeaufortScale(const double windSpeeds[], int size) {

int minBeaufort = 12;

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

int beaufort = getBeaufortScale(windSpeeds[i]);

if (beaufort < minBeaufort) {

minBeaufort = beaufort;

}

}

return minBeaufort;

}

int analyzeBinary(int N) {

bitset<32> bits(N);

return ((N >> 9) & 1) ? 32 - bits.count() : bits.count();

}

void s\_calculation() {

cout << "Виклик функції s\_calculation()" << endl;

}

int main() {

string input;

while (true) {

cout << "Введіть символ (d/g/h/j, k/к для виходу): ";

getline(cin, input);

if (input.empty()) continue;

char cmd = tolower((unsigned char)input[0]);

switch (cmd) {

case 'd':

s\_calculation();

break;

case 'g': {

double kilowatts;

cout << "Введіть кількість кВт·год: ";

cin >> kilowatts;

cin.ignore();

cout << "Сума до сплати: " << calculateElectricityBill(kilowatts) << " грн" << endl;

break;

}

case 'h': {

double windSpeeds[6];

cout << "Введіть 6 значень сили вітру (м/с): ";

for (int i = 0; i < 6; ++i) cin >> windSpeeds[i];

cin.ignore();

cout << "Мінімальний бал за шкалою Бофорта: " << getMinBeaufortScale(windSpeeds, 6) << endl;

break;

}

case 'j': {

int N;

cout << "Введіть число N (0 до 500700): ";

cin >> N;

cin.ignore();

if (N >= 0 && N <= 500700) {

cout << "Результат аналізу: " << analyzeBinary(N) << endl;

} else {

cout << "Помилка: число поза діапазоном." << endl;

}

break;

}

case 'k':

case 'к':

cout << "Вихід з програми." << endl;

return 0;

default:

cout << "\aНеправильне введення символу." << endl;

}

}

}

**Висновок:**

У ході виконання лабораторної роботи я навчився реалізовувати програмні модулі для розгалужених та ітераційних обчислювальних процесів. Я здобув практичні навички використання операторів мови програмування С/С++, зокрема арифметичних, логічних, побітових, а також умовних операторів, операторів циклів і вибору. Особливу увагу було приділено технології модульного програмування: я навчився створювати та використовувати заголовкові файли й статичні бібліотеки у проєкті в середовищі Code::Blocks. Завдяки виконанню завдань 9.1–9.4 я закріпив навички організації коду у вигляді бібліотек та розробки кросплатформового програмного забезпечення з використанням функцій власної бібліотеки.