Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Звіт

з лабораторної роботи № 4

з дисципліни “Базові методології та технології програмування”

на тему

“Програмна реалізація лінійних алгоритмічних конструкцій”

Виконав студент

академічної групи КІ-20

Микитенко Д. Ю.

Перевірив доцент

Доренський О. П.

Кропивницький-2020

Мета роботи: полягає у набутті ґрунтовних вмінь та практичних навичок розроблення лінійних алгоритмів розв’язування обчислювальних задач, їх формального представлення мовою програмування С (ISO/IEC 9899:2018) та реалізації програмного забезпечення у вільному кросплатформовому інтегрованому середовищі Code::Blocks (GNU GCC Compiler).

Завдання:

1. Створити додаток з вихідного коду програми лістинга 4.1.
2. Алгоритм 4.1 формально представити мовою програмування C.
3. Реалізувати програмне забезпечення розв’язування задачі 4.1.

Варіант 6

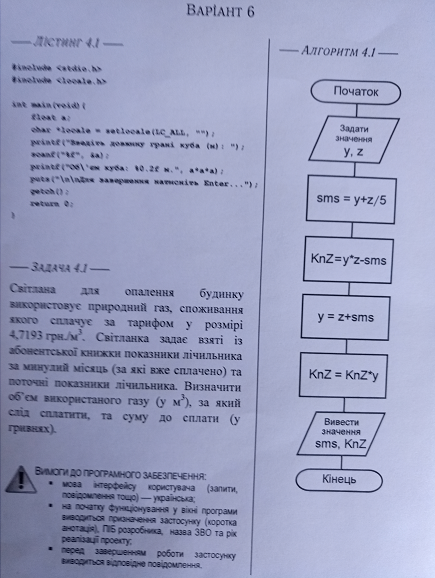


Рисунок 1 - Варіант роботи

1. Згідно з п.3-5 Порядку були створені деки: prj, Software, TestSuite.

Лістинг 4-1 був записаний у IDE Code::Blocks у папці prj.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main(void)

{

char\* locale = setlocale(LC\_ALL, "");

float a;

printf("Введіть довжину грані куба (м):");

scanf ("%f", &a);

printf ("Об'єм куба: %0.2f м", a\*a\*a);

puts ("\n\n Для завершення натисніть Enter...");

getch();

return 0;

}

Лістинг успішно компілюється.

1. Aлгоритм 4-1був записаний у IDE Code::Blocks.

За допомогою аналізу був сформований перелік змінних(вхідні та вихідні значення).

Вихідні дані:

1) sms - тип float(дійсне число);

2) KnZ - тип float(дійсне число).

Вхідні дані: 1) y – тип float(дійсне число);

2) z – тип float(дійсне число).

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main()

{

char \*locale = setlocale(LC\_ALL, "");

float y,z,sms,KnZ;

printf("Введіть значення y\n");

scanf("%f", &y);

printf("Введіть значення z\n");

scanf ("%f",&z);

sms = y+z/5;

KnZ = y\*z-sms;

y = z+sms;

KnZ = KnZ\*y;

printf("%2.f, %2.f", sms, KnZ);

getch();

}

1. Задача 4-1 була записана мовою програмування С у IDE Code::Blocks.

За допомогою аналізу був сформований перелік змінних(вхідні та вихідні значення).

Вихідні дані:

1) vol –дійсне число – відповідає за суму до сплати;

2) dig - дійсне число – відповідає за об’єм газу до сплати.

Вхідні дані:

1) gasval – дійсне число – константа, що відповідає за тариф газу;

2) last –дійсне число – відповідає за минулі показники лічильника;

3) cur –дійсне число – відповідає за поточні показники лічильника.

Математична модель алгоритму:

1. vol = (cur - last) \* gasvol
2. dig = cur - last

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main()

{

char \*locale = setlocale(LC\_ALL,"");

printf("Розробник - Микитенко Даниїл Юрiйович, студент ЦНТУ. 2020 рiк.\nПризначення застосунку - визначити об'єм використанного газу, за який слiд сплатити, та суму до сплати.\n\n");

double gasval = 4.7193;

double last, cur, vol;

printf("Введiть показники лiчильника за минулий мiсяць:\n");

scanf("%lf", &last);

printf("Введiть поточнi показники лiчильника:\n");

scanf("%lf", &cur);

vol = (cur-last)\*gasval;

double dig = cur-last;

printf("Неоплачено %2.lf кубiчних метри. До сплати: %2.lf грн\n\n", dig, vol);

}

Код компілюється, умова виконується. Усі тест-сьюти були успішно перевірені.

ПЗ було сформульовано у вербальному вигляді:

1. Ввести last.
2. Ввести cur.
3. Отримати результат.

Висновок: у ході виконання лабораторної роботи №4 треба було записати лістинг й скомпілювати його, записати алгоритм структурного вигляду мовою програмування С, реалізувати програмне забезпечення розв’язування заданої задачі.

Щоб виконати перше завдання достатньо було отримати листок з варіантом роботи у викладача, проаналізувати лістинг, задокументувати вихідні й вхідні значення, їх типи й побудувати математичну модель, після чого створити проєкт консольного застосування у IDE Code::Blocks й безпосередньо записати сам лістинг 4-1. Під час виконання завдання труднощів не було виявлено й проєкт компілювався. Після цього .exe файл першого завдання був перенесений до теки Software, що була створена до цього на подобі до лабораторної роботи №3.

Для виконання другого завдання потрібно було проаналізувати блок-схему алгоритму й чітко з’ясувати: які можуть бути вхідні та вихідні дані для їх типізації, записати математичну модель. Після уважного аналізу потрібно було записати алгоритм 4-1 мовою програмування С у середовищі IDE Code::Blocks. Під час виконання другого завдання складнощів також не було виявлено. .exe файл був перенесений до теки Software.

Задля виконання третього завдання потрібно було уважно проаналізувати дану задачу, задокументувати вихідні й вхідні значення, їх типи й побудувати математичну модель. Після цього був створений шаблон програмного забезпечення до задачі 4-1, адже ПЗ не було протестовано. Потім було створено 10 тест-сьютів для умови задачі й було перевірено кожен з них для нашого програмного забезпечення, після чого виявилося

Підбиваючи підсумки цієї лабораторної роботи, були закріплені практичні вміння праці з тестуванням ПЗ, закріпленні знання з теми структурного вигляду алгоритму (блок-схеми). Також були здобуті практичні вміння з розробкою ПЗ для окремої задачі та обмеження вхідних та вихідних значень.