

Лабораторна робота № 4

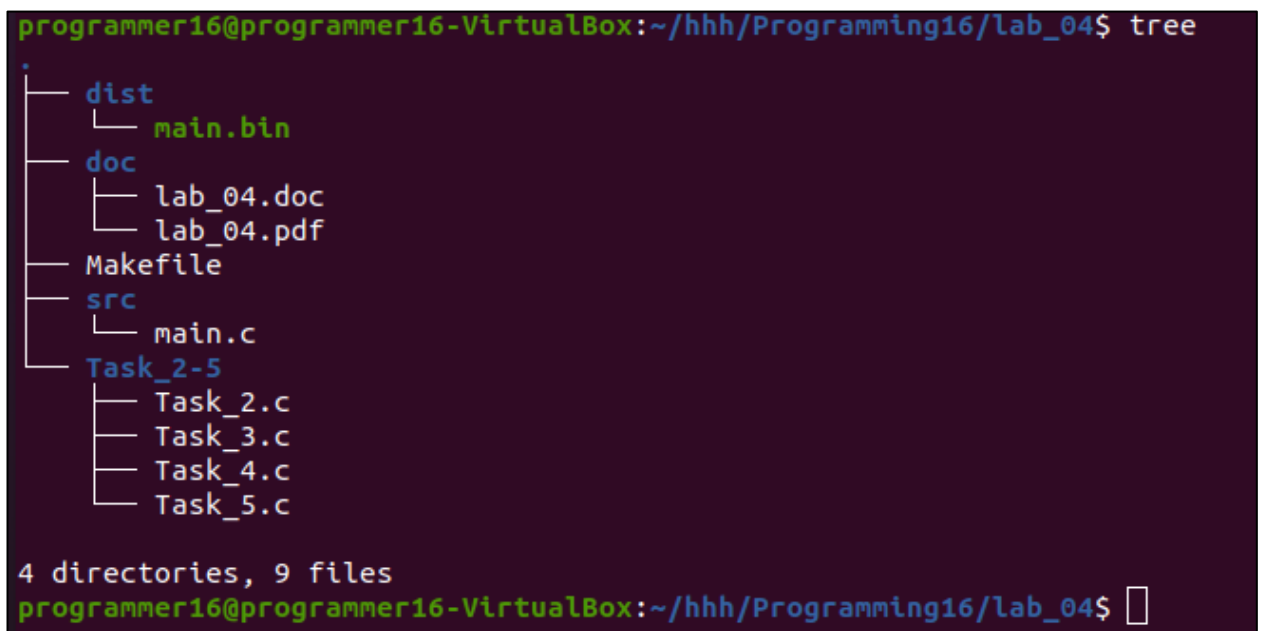
Тема: Розробка лінійних програм

Розробник: Студент 1-го курсу, Михневич Владислав Вікторович,
групи КІТ-120а.

Загальне завдання: За заданим радіусом r та командою ('l', 's' або 'v') користувача обчислити:

- Довжину окружності, якщо команда - 'l';
- Площу кола, якщо команда – 's';
- Об'єм кулі, якщо команда – 'v'.

1. Створимо у папці свого репозиторію папку lab_04, в яку будемо додавати нові файли (Рис. 1).



```
programmer16@programmer16-VirtualBox:~/hhh/Programming16/lab_04$ tree
.
├── dist
│   └── main.bin
├── doc
│   ├── lab_04.doc
│   └── lab_04.pdf
├── Makefile
├── src
│   └── main.c
└── Task_2-5
    ├── Task_2.c
    ├── Task_3.c
    ├── Task_4.c
    └── Task_5.c

4 directories, 9 files
programmer16@programmer16-VirtualBox:~/hhh/Programming16/lab_04$
```

Рисунок 1 – Структура проекту

2. Перейдемо до написання коду в папо та творимо тіло мови C (Рис. 2).

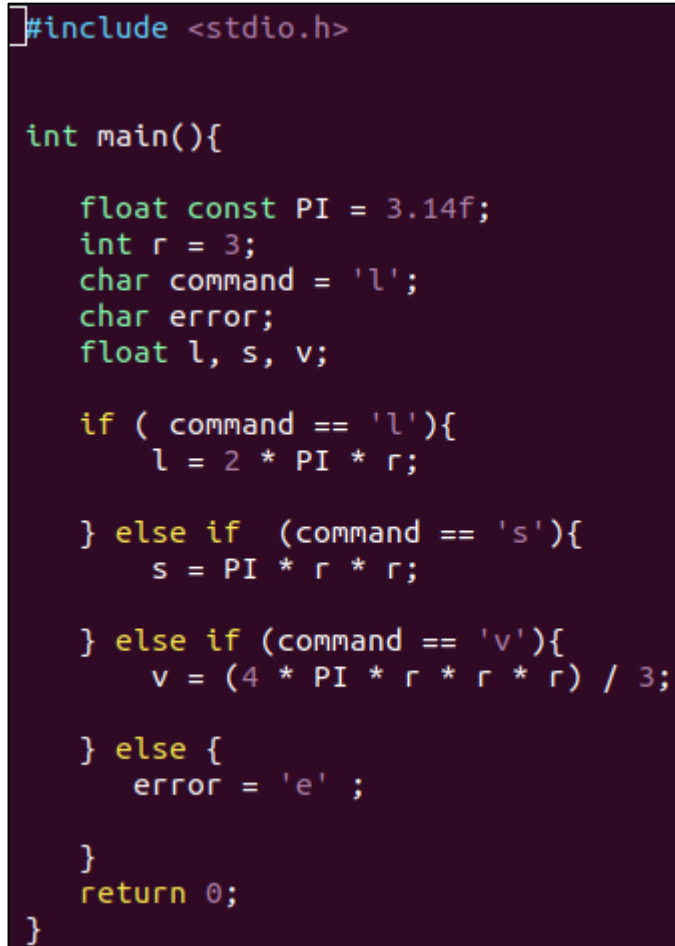
```
int main(){
    return 0;
}
```

3. Задамо значення змінних та типи даних (Рис. 2).

```
float const PI = 3.14f;
int r = 3;
char command = 'l';
char error;
float l, s, v;
```

4. Запишемо формули для обчислення довжини, площі та об'єму кола в умовному операторі else if (Рис. 2).

```
if (command == 'l'){  
    l = 2 * PI * r;  
} else if (command == 's') {  
    s = PI * r * r;  
} else if (command == 'v') {  
    v = (4 * PI * r * r * r) / 3;  
} else {  
    error = 'e';  
}
```

A screenshot of a code editor with a dark purple background. The code is written in C and implements a program to calculate the circumference, area, and volume of a circle based on a command. The code includes a header file, defines constants and variables, and uses an if-else-if structure for the logic. The code is as follows:

```
#include <stdio.h>  
  
int main(){  
  
    float const PI = 3.14f;  
    int r = 3;  
    char command = 'l';  
    char error;  
    float l, s, v;  
  
    if ( command == 'l'){  
        l = 2 * PI * r;  
  
    } else if (command == 's'){  
        s = PI * r * r;  
  
    } else if (command == 'v'){  
        v = (4 * PI * r * r * r) / 3;  
  
    } else {  
        error = 'e' ;  
  
    }  
    return 0;  
}
```

Рисунок 2 – Код в nano

5. Скомпілюємо програму за допомогою команди:

```
gcc -g main.c -o main.bin.
```

Та побачимо, що компіляція пройшла успішно.

6. Створимо файл Makefile (Рис. 3).

```
GNU nano 4.8      Makefile
targets = main.bin
CC = gcc
C_OPTS = -std=gnu11 -g -Wall -Wextra -Werror -Wformat-s>

all: clean prep compile run

clean:
    rm -rf dist
prep:
    mkdir dist
compile: main.bin
main.bin: src/main.c
    $(CC) $(C_OPTS) $< -o ./dist/$@
run: clean prep compile
    ./dist/main.bin

^G  Помощь      ^O  Записать      ^W  Поиск        ^K  Вырезать
^X  Выход      ^R  ЧитФайл      ^\  Замена      ^U  Paste Text
```

Рисунок 3 – Структура створення Makefile

7. Запустимо програму та перевіримо за допомогою Nemiver та побачимо що все гаразд (Рис. 4).
8. Точка входу знаходиться на 4 строчці та як за умовою з нас не вимагається створення інших функцій (Рис. 4).

main.c

```

1  #include <stdio.h>
2
3
4  int main(){
5
6      float const PI = 3.14f;
7      int r = 3;
8      char command = 'l';
9      char error;
10     float l, s, v;
11
12     if ( command == 'l'){
13         l = 2 * PI * r;
14
15     } else if (command == 's'){
16         s = PI * r * r;
17
18     } else if (command == 'v'){
19         v = (4 * PI * r * r * r) / 3;
20
21     } else {
22         error = 'e' ;
23
24     }
25     return 0;
26 }
```

ID по	Кадр	Функция	Переменная	Значение	Тип
1	0	main	Локальные переменные		
			PI	3.1400001	const float
			r	3	int
			command	108 'l'	char
			error	85 'U'	char
			l	18.8400002	float
			s	0	float
			v	4.59163468e-41	float
			Параметры функции		

Рисунок 4 – Перевірка у Nemiver

9. Після того як ми розробили та перевірили нашу програму можемо додати її до нашого репозиторію на Github. За допомогою послідовності команд:

```
git add .
git commit -m "Назва коміту"
git push
```

Висновок: На цій лабораторній роботі ми навчилися розробляти програми, що розгладжуються, ознайомилися з типом даних `char`, дізналися як працює умовний оператор `if` та оператор `switch` та використали їх на практиці, дізналися про роботу оператора `break`.