

Основи програмування 2. Модульне програмування

Міністерство освіти і науки України
Національний технічний університет України “Київський політехнічний
інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни
«Основи програмування 2. Модульне програмування»

«Класи та об'єкти»

Варіант 29

Виконав студент

ПІ-15 Рибалка Ілля Сергійович
(шифр, прізвище, ім'я, по батькові)

Перевірів

Всечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 2

Класи та об'єкти

Мета - вивчити механізм створення і використання класів та об'єктів.

Індивідуальне завдання

2. Вимоги до програми

- 1) Реалізація коду відповідно до модульного підходу.
- 2) Виведення усіх вихідних, проміжних та результуючих даних.

Варіант 29

29. Розробити клас, який є абстракцією тексту та підтримує операції додавання рядка до тексту і визначення частоти появи у тексті заданого символу (кількість символів/загальна кількість символів). Створити масив об'єктів даного класу. Доповнити декілька текстів новими рядками. Визначити символ, який найчастіше зустрічається у текстах.

Розв'язання

Код

main.cpp

```
#include "func.h"

int main()
{
    int len, num;

    char ch;

    Text *texts = inputtexts(&len);

    addtotexts(texts, len);

    print(texts, len);

    countintext(texts, len);
```

Основи програмування 2. Модульне програмування

```
ch = findmax(texts, len, &num);

    std::cout << "Найчастіше зустрічається символ '" << ch << "' - " << num <<
std::endl;

    delete []texts;

    return 0;

}
```

func.h

```
#pragma once

#include "Text.h"

Text *inputtexts(int *); // Введення текстів з консолі

void addtotexts(Text [], int); // Введення додаткового рядку до тексту

void countintext(Text [], int); // Підрахунок введеного символу в тексті

char findmax(Text [], int, int *); // Знаходження символу що більше всього повторюється
у всіх текстах

int find(std::vector<char>, char); // Знаходження індексу елемента у векторі символів

int max(std::vector<int>); // Знаходження індексу максимального елемента у векторі
чисел

void print(Text [], int); // Виведення текстів

bool ask(std::string); // Виведення запиту в консоль
```

func.cpp

```
#include "func.h"

Text *inputtexts(int *len)

{

    int n;

    std::cout << "Введіть кількість текстів - ";

    std::cin >> n;
```

```
if (n<=0)

{

    std::cout << "Введіть натуральне число - ";

    std::cin >> n;

}

*len = n;

Text *texts = new Text[n];

std::string row;

for (int i = 0; i < n; i++)

{

    if (ask("Вводити текст вручну?"))

    {

        std::cout << "Введіть текст:" << std::endl;

        std::cin.ignore();

        std::getline(std::cin, row);

        Text t(row);

        std::getline(std::cin, row);

        while(row.find(1) == std::string::npos) // Ознака кінця введення Ctrl+A

        {

            t.addRow(row);

            std::getline(std::cin, row);

        }

        texts[i] = t;

    }

    else
```

```

        {

            Text t;

            texts[i] = t;

        }

    }

    return texts;
}

void addtotexts(Text texts[], int len)
{

    std::string row;

    for (int i = 0; i < len; i++)

    {

        if (ask("Додати рядок до тексту?"))

        {

            std::cout << "Введіть рядок:" << std::endl;

            std::cin.ignore();

            std::getline(std::cin, row);

            texts[i].addRow(row);

        }

    }

}

void countintext(Text texts[], int len)
{

```

```

char ch;

for (int i = 0; i < len; i++)

{

    if (ask("Ввести свій символ для підрахунку?"))

    {

        std::cout << "Введіть символ - ";

        std::cin >> ch;

        std::cout << "Кількість входження символу " << ch << " в текст №" << i+1 <<
" - " << texts[i].count(ch) << std::endl;

    }

    else std::cout << "Кількість символів в тексті №" << i+1 << " - " <<
texts[i].count() << std::endl;

}

}

char findmax(Text texts[], int len, int *num)

{

    std::vector<int> n;

    std::vector<char> temp;

    int tn, indx;

    std::vector<char> ch;

    std::string txt;

    for (int i = 0; i < len; i++)

    {

        txt = texts[i].GetText();

        for (int j = 0; j < txt.size(); j++)
    
```

```

{

    if(txt[j]!=' ')

    {

        indx = find(ch, txt[j]);

        if(indx == -1)

        {

            tn = texts[i].count(txt[j]);

            n.push_back(tn);

            ch.push_back(txt[j]);

            temp.push_back(txt[j]);

        }

        else if (find(temp, txt[j])== -1)

        {

            tn = texts[i].count(txt[j]);

            n[indx] += tn;

            temp.push_back(txt[j]);

        }

    }

    temp.clear();

}

tn = max(n);

*num = n[tn];

return ch[tn];

}

```

```
int find(std::vector<char> chs, char ch)

{

    for (int i = 0; i < chs.size(); i++) if(chs[i]==ch) return i;

    return -1;

}
```

```
int max(std::vector<int> nums)

{

    int indx = 0;

    for (int i = 0; i < nums.size(); i++)

    {

        if(nums[indx] < nums[i]) indx = i;

    }

    return indx;

}
```

```
void print(Text texts[], int len)

{

    for (int i = 0; i < len; i++)

    {

        std::cout << "Текст №" << i+1 << ":" << std::endl;

        texts[i].printtxt();

    }

}
```



```
bool ask(std::string str)
{
    char ans;

    std::cout << str << " Y/N ";

    std::cin >> ans;

    while (ans != 'Y' && ans != 'y' && ans != 'N' && ans != 'n')
    {

        std::cout << "Y/N ";

        std::cin >> ans;

    }

    if (ans == 'Y' || ans == 'y') return true;

    else return false;
}
```

Text.h

```
#pragma once

#include <iostream>

#include <string>

#include <vector>

class Text
{
    std::vector<std::string> t;

public:

    Text(); // Конструктор для створення тексту без введенного рядка
}
```

```
Text(std::string); // Конструктор для створення тексту за введеним рядком

void addRow(std::string); // Додавання рядка до тексту

std::string GetText(); // Повертає текст у вигляді строки

int count(); // Підрахунок усіх символів в тексті

int count(char); // Підрахунок введеного символу в тексті

void printtxt(); // Виведення тексту

};
```

Text.cpp

```
#include "Text.h"

Text::Text()
{
    this->t.push_back("Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do  

    eiusmod tempor incididunt ut labore et dolore magna aliqua.");
}

Text::Text(std::string t)
{
    this->t.push_back(t);
}

void Text::addRow(std::string row)
{
    this->t.push_back(row);
}

std::string Text::GetText()
{
    std::string txt;
    for (int i = 0; i < this->t.size(); i++)
    {
        txt += this->t[i];
    }
    return txt;
}

int Text::count()
{
}
```

```

    int res = 0;
    for (int i = 0; i < this->t.size(); i++)
    {
        for (int j = 0; j < this->t[i].size(); j++)
        {
            if (this->t[i][j] != ' ') res++;
        }
    }
    return res;
}

int Text::count(char ch)
{
    int res = 0;
    for (int i = 0; i < this->t.size(); i++)
    {
        for (int j = 0; j < this->t[i].size(); j++)
        {
            if (this->t[i][j] == ch) res++;
        }
    }
    return res;
}

void Text::printtxt()
{
    for (int i = 0; i < this->t.size(); i++)
    {
        std::cout << this->t[i] << std::endl;
    }
}

```

Тестування

Основи програмування 2. Модульне програмування

```
Введіть кількість текстів - 2
Вводити текст вручну? Y/N Y
Введіть текст:
eee
ee
e
^A
Вводити текст вручну? Y/N n
Додати рядок до тексту? Y/N Y
Введіть рядок:
eee
Додати рядок до тексту? Y/N N
Текст №1:
eee
ee
e
eee
Текст №2:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ввести свій символ для підрахунку? Y/N N
Кількість символів в тексті №1 - 9
Ввести свій символ для підрахунку? Y/N Y
Введіть символ - e
Кількість входження символу e в текст №2 - 11
Найчастіше зустрічається символ 'e' - 20
```

Висновок

Я дослідив механізм створення і використання класів та об'єктів. В ході роботи над лабораторною роботою було створено програму для внесення і обробки тексту.