Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет комп’ютерних систем та автоматики

Кафедра КСУ

Лабораторна робота №7

з дисципліни: «Компютерні технології та програмування»

Тема: «Класи. Перевантаження функцій»

Виконав

студент групи 2АКІТ-20б

Н. Бондар

Перевірив

к.т.н., доц.кафедри КСУ

Севастьянов В. М.

м.Вінниця, ВНТУ 2021

**Лабораторна робота №7**

**Тема: «Класи. Перевантаження функцій»**

**Варіант 1.**

**Завдання :**

1. Модернізувати клас з попередньої лабораторної роботи відповідно до варіанту.

2. Написати програму, яка буде демонструвати роботу з функціями-членами класу.

Для демонстрації роботи програма повинна містити меню. В програмі використати

один екземпляр класу для кожного конструктора.

Варіант 1. Клас «Залікова книжка»

. Закриті дані-члени класу:

- Прізвище студента;

- Номер залікової книжки.

Відкриті функції-члени класу:

- Ініціалізація (введення через аргументи функції інформації у дані-члени класу); -

Виведення на екран інформації збереженої в даних-членах.

2.Створити структуру «Дисципліна»:

- Дисципліна;

- Оцінка;

- Прізвище викладача.

До закритих даних-членів добавити:

- назва факультету (статичний елемент);

До відкритих функцій-членів добавити:

- функцію яка виводить на екран назву факультету (статичний елемент);

Створити два конструктори, які будуть зчитувати дані для заповнення масиву

структур та даних-членів класу з текстового файлу. Перший конструктор без параметрів.

В другий конструктор через параметр зчитується назва текстового файлу.

У конструкторі створити динамічний масив, з елементами типу «Дисципліна».

Кількість елементів задається з файлу.

Деструктор має виводити на екран вміст об’єкту типу «Залікова книжка».

Приклад текстового файлу (zk.txt):

2

Васаженко

А-111111

АМтаПР

3

Яценюк

Фізика

3

Ющенко

**Хід роботи :**

**1. Ретельно вивчаю поставлене передімною завдання.**

**2. Створюю програму на мові С згідно варіанту використавши середовище програмування Dev-C++ 4.0:**

**3. Відкомпільовую та відлагоджую програму.**

**4. Розробляю набір тестів і перевіряю роботу програми на них.**

**5. Відповідаю на контрольні запитання.**

**6. Роблю висновки.**

**1. Розписуюю структуру Discipline:**

struct Discipline {

public:

char \* discipline;

int mark;

char \* teacher\_last\_name;

private:

const char \* faculty = "FKSA";

public:

void show\_faculty() {

std::cout << "Faculty : " << faculty << std::endl;

}

Discipline() {

discipline = NULL;

mark = 0;

teacher\_last\_name = NULL;

}

Discipline(char \* discipline, int mark, char \* teacher\_last\_name) {

//std::cout << "Create an object !!!\n";

this->discipline = new char[lenght(discipline)];

this->teacher\_last\_name = new char[lenght(teacher\_last\_name)];

copy(discipline, this->discipline);

copy(teacher\_last\_name, this->teacher\_last\_name);

this->mark = mark;

}

~Discipline() {

//std::cout << "Delete a object !!!\n";

delete [] discipline;

delete [] teacher\_last\_name;

}

};

**2. Розписую клас student\_gradebook :**

class student\_gradebook {

private:

char \* last\_name;

char \* gradebook;

int subjects\_number;

Discipline \* discipline;

public:

student\_gradebook(char \* last\_name, char \* gradebook) {

//std::cout << "Create an object !!!\n";

this->last\_name = new char[lenght(last\_name)];

this->gradebook = new char[lenght(gradebook)];

copy(last\_name, this->last\_name);

copy(gradebook, this->gradebook);

}

student\_gradebook(char \* file\_name) {

char char\_temp[100];

int int\_temp;

FILE \* file = fopen(file\_name, "r");

fscanf(file, "%i\n", &subjects\_number);

fscanf(file, "%s\n", char\_temp);

last\_name = new char[lenght(char\_temp)];

copy(char\_temp, this->last\_name);

fscanf(file, "%s\n", char\_temp);

gradebook = new char[lenght(char\_temp)];

copy(char\_temp, this->gradebook);

this->discipline = new Discipline[subjects\_number];

// std::cout << "\nSubjects\_number : " << subjects\_number << "\nStudent's last name : " << last\_name << "\nStudent's gradebook : " << gradebook << "\n\n";

for (int i = 0; i < subjects\_number; i++) {

fscanf(file, "%s\n", char\_temp);

// std::cout << "Subject : " << char\_temp << '\n';

discipline[i].discipline = new char[strlen(char\_temp)];

strcpy(discipline[i].discipline, char\_temp);

fscanf(file, "%i\n", &int\_temp);

// std::cout << "Mark : " << int\_temp << '\n';

discipline[i].mark = int\_temp;

fscanf(file, "%s\n", char\_temp);

// std::cout << "Teacher : " << char\_temp << "\n\n";

discipline[i].teacher\_last\_name = new char[strlen(char\_temp)];

strcpy(discipline[i].teacher\_last\_name, char\_temp);

}

fclose(file);

}

~student\_gradebook() {

//std::cout << "Delete a object !!!\n";

std::cout << "Student\'s last name : " << last\_name << "\nGradebook number : " << gradebook << std::endl << std::endl;

for (int i = 0; i < subjects\_number; i++) {

std::cout << "Subject : " << discipline[i].discipline << '\n';

std::cout << "Mark : " << discipline[i].mark << '\n';

std::cout << "Teacher : " << discipline[i].teacher\_last\_name << "\n\n";

}

delete [] last\_name;

delete [] gradebook;

}

};

**3. Розписую головну функцію :**

int main() {

// student\_gradebook gradebook("Ivanov V.V.", "123456");

student\_gradebook gradebook("zk.txt");

// Discipline discipline("Phisics", 5, "Ivanov");

}

**4. Розписую допоміжні функції :**

**Лістинг коду :**

/\*

Варіант 1. Клас «Залікова книжка»

. Закриті дані-члени класу:

- Прізвище студента;

- Номер залікової книжки.

Відкриті функції-члени класу:

- Ініціалізація (введення через аргументи функції інформації у дані-члени класу); -

Виведення на екран інформації збереженої в даних-членах.

2.Створити структуру «Дисципліна»:

- Дисципліна;

- Оцінка;

- Прізвище викладача.

До закритих даних-членів добавити:

- назва факультету (статичний елемент);

До відкритих функцій-членів добавити:

- функцію яка виводить на екран назву факультету (статичний елемент);

Створити два конструктори, які будуть зчитувати дані для заповнення масиву

структур та даних-членів класу з текстового файлу. Перший конструктор без параметрів.

В другий конструктор через параметр зчитується назва текстового файлу.

У конструкторі створити динамічний масив, з елементами типу «Дисципліна».

Кількість елементів задається з файлу.

Деструктор має виводити на екран вміст об’єкту типу «Залікова книжка».

Приклад текстового файлу (zk.txt):

2

Васаженко

А-111111

АМтаПР

3

Яценюк

Фізика

3

Ющенко

\*/

#include<iostream>

#include<cstdlib>

#include<cstring>

void copy(char \* text\_1, char \* text\_2);

int lenght(char \* text);

struct Discipline {

public:

char \* discipline;

int mark;

char \* teacher\_last\_name;

private:

const char \* faculty = "FKSA";

public:

void show\_faculty() {

std::cout << "Faculty : " << faculty << std::endl;

}

Discipline() {

discipline = NULL;

mark = 0;

teacher\_last\_name = NULL;

}

Discipline(char \* discipline, int mark, char \* teacher\_last\_name) {

//std::cout << "Create an object !!!\n";

this->discipline = new char[lenght(discipline)];

this->teacher\_last\_name = new char[lenght(teacher\_last\_name)];

copy(discipline, this->discipline);

copy(teacher\_last\_name, this->teacher\_last\_name);

this->mark = mark;

}

~Discipline() {

//std::cout << "Delete a object !!!\n";

delete [] discipline;

delete [] teacher\_last\_name;

}

};

class student\_gradebook {

private:

char \* last\_name;

char \* gradebook;

int subjects\_number;

Discipline \* discipline;

public:

student\_gradebook(char \* last\_name, char \* gradebook) {

//std::cout << "Create an object !!!\n";

this->last\_name = new char[lenght(last\_name)];

this->gradebook = new char[lenght(gradebook)];

copy(last\_name, this->last\_name);

copy(gradebook, this->gradebook);

}

student\_gradebook(char \* file\_name) {

char char\_temp[100];

int int\_temp;

FILE \* file = fopen(file\_name, "r");

fscanf(file, "%i\n", &subjects\_number);

fscanf(file, "%s\n", char\_temp);

last\_name = new char[lenght(char\_temp)];

copy(char\_temp, this->last\_name);

fscanf(file, "%s\n", char\_temp);

gradebook = new char[lenght(char\_temp)];

copy(char\_temp, this->gradebook);

this->discipline = new Discipline[subjects\_number];

// std::cout << "\nSubjects\_number : " << subjects\_number << "\nStudent's last name : " << last\_name << "\nStudent's gradebook : " << gradebook << "\n\n";

for (int i = 0; i < subjects\_number; i++) {

fscanf(file, "%s\n", char\_temp);

// std::cout << "Subject : " << char\_temp << '\n';

discipline[i].discipline = new char[strlen(char\_temp)];

strcpy(discipline[i].discipline, char\_temp);

fscanf(file, "%i\n", &int\_temp);

// std::cout << "Mark : " << int\_temp << '\n';

discipline[i].mark = int\_temp;

fscanf(file, "%s\n", char\_temp);

// std::cout << "Teacher : " << char\_temp << "\n\n";

discipline[i].teacher\_last\_name = new char[strlen(char\_temp)];

strcpy(discipline[i].teacher\_last\_name, char\_temp);

}

fclose(file);

}

~student\_gradebook() {

//std::cout << "Delete a object !!!\n";

std::cout << "Student\'s last name : " << last\_name << "\nGradebook number : " << gradebook << std::endl << std::endl;

for (int i = 0; i < subjects\_number; i++) {

std::cout << "Subject : " << discipline[i].discipline << '\n';

std::cout << "Mark : " << discipline[i].mark << '\n';

std::cout << "Teacher : " << discipline[i].teacher\_last\_name << "\n\n";

}

delete [] last\_name;

delete [] gradebook;

}

};

int main() {

// student\_gradebook gradebook("Ivanov V.V.", "123456");

student\_gradebook gradebook("zk.txt");

// Discipline discipline("Phisics", 5, "Ivanov");

}

void copy(char \* text\_1, char \* text\_2) {

int i = 0;

for ( ; text\_1[i] != '\0'; i++) {

text\_2[i] = text\_1[i];

}

for ( ; text\_2[i] != '\0'; i++) {

text\_2[i] = '\0';

}

}

int lenght(char \* text) {

int i = 0;

for ( ; text[i] != '\0'; i++) {}

return i;

}

**Висновок :** виконуючи дану лабораторно роботу, я навчився працювати з класа та структурами для створення масивів об’ектів