Министерство образование Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Информационно-технологический факультет

Кафедра технологий программирования

Лабораторная работа №1 по теме «Классы и объекты в C++»

По дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Выполнил: Храповицкая К.О.

Группа 19-ИТ-1.

Проверил: Васильева Д. М.

Новополоцк

2021 г.

**Вариант 7**

**Цель работы**: получить практические навыки реализации классов на С++.

**Задание 1. Описание членов-данных пользовательских классов.**

Согласно моему варианту необходимо описать пользовательских класс «Библиотека», который содержит:

1. Название книги – char\*
2. Автора–char\*
3. Цена – float

Дополнительно должны быть реализованы:

1. Метод-аксессоры для каждого из полей
2. Конструктор без параметров
3. Конструктор с параметрами
4. Конструктор копирования

**Конструктор** — это особый тип метода класса, который автоматически вызывается при создании объекта этого же класса. Конструкторы обычно используются для инициализации переменных-членов класса значениями, которые предоставлены по умолчанию/пользователем, или для выполнения любых шагов настройки, необходимых для используемого класса (например, открыть определенный файл или базу данных).

Конструктор, который не имеет параметров (или содержит параметры, которые все имеют значения по умолчанию), называется **конструктором по умолчанию**. Он вызывается, если пользователем не указаны значения для инициализации.

Хотя конструктор по умолчанию отлично подходит для обеспечения инициализации наших классов значениями по умолчанию, часто может быть нужно, чтобы экземпляры нашего класса имели определенные значения, которые мы предоставим позже. К счастью, **конструкторы также могут быть объявлены с параметрами**.

Подобно обычным переменным, классы также можно инициализировать, используя копирующую инициализацию, с использованием конструктора копирования.

**Функция доступа** — это короткая открытая функция, задачей которой является получение или изменение значения закрытой переменной-члена класса.

Функции доступа обычно бывают двух типов:

**Геттеры**— это функции, которые возвращают значения закрытых переменных-членов класса;

**Сеттеры** — это функции, которые позволяют присваивать значения закрытым переменным-членам класса.

Тогда, используя данную теорию, мы получаем следующий класс:

1. class Library
2. {
3. private:
4. // поля класса
5. char name[LNAME];
6. char aname[ANAME];
7. float coust;
8. public:
9. // конструкторы класса по умолчанию
10. Library();
11. // конструктор класса с параметрами
12. Library(char\*, char\*, float);
13. // конструктор копирования
14. Library(const Library&);
15. // деструктор
16. ~Library();
17. // геттеры
18. char\* getName();
19. char\* getAname();
20. float getCoust();
21. // сеттеры
22. void setName(char\*);
23. void setAname(char\*);
24. void setCoust(float);
25. void set(char\*, char\*, float);
26. // методы вывода информации о объекте
27. void show();
28. };

Реализация конструкторов и деструктора:

1. Library::Library() {
2. //присваивание по умолчанию
3. char NAME[LNAME] = "Default name";
4. char ANAME[ANAME] = "Default name";
5. strcpy\_s(name, NAME);
6. strcpy\_s(aname,ANAME);
7. coust = 0;
8. //вывод
9. show();
10. cout << "Object above is created" << endl;
11. }
12. Library::Library(char\* NAME, char\* AUTHOR\_NAME, float COUST) {
13. //присваивание переданных значений
14. strcpy\_s(name, NAME);
15. strcpy\_s(aname, AUTHOR\_NAME);
16. coust = COUST;
17. //вывод
18. show();
19. cout << "Object above is created" << endl;
20. }
21. Library::Library(const Library& LIBRARY) {
22. //копирование
23. strcpy\_s(name, LIBRARY.name);
24. strcpy\_s(aname, LIBRARY.aname);
25. coust = LIBRARY.coust;
26. //вывод
27. show();
28. cout << "Object above is created" << endl;
29. }
30. Library::~Library() {
31. //удаление
32. show();//вывод
33. cout << "Object above is deleted" << endl;
34. }

Листинг программы с использование функции-указателя:

1. int main()
2. {
3. // создание инстанса класса и присваивание указателю
4. // в данном случае будет вызван конструктор по умолчанию
5. Library\* library = new Library();
6. // инициализация переменных
7. char name[25] = "Book name";
8. char aname[25] = "Author name ";
9. float coust = 1.1;
10. library->set(name, aname, coust);
11. // использование функции указателя
12. void (Library:: \* ptLibrary) () = &Library::show;
13. (\*library.\*ptLibrary)();
14. // удаление
15. delete library;
16. }