Лабораторная работа №1
по теме: "Классы. Конструкторы и деструкторы"

<u>Задание 1.</u> Создание класса

1. Создайте класс ***Stock***, предназначенный для работы с пакетом акций компаний. Необходимо хранить информацию о наименовании компании, количестве акций в пакете, цене каждой акции, общей стоимости всех акций (для подсчета общей стоимости всех акция использовать закрытую функцию ***set_tot()***).

2. В классе должны быть определены следующие функции:
a) Функция ***acquire*** выполняет обработку первоначального вклада конкретной компании, запоминая информацию о наименовании компании, количестве акций в пакете, цене каждой акции данного пакета и вычисляя общую стоимость пакета акций с помощью функции ***set_tot()***;
b) Функция ***buy()*** предназначена для приобретения дополнительных акций;
c) Функция ***sell()*** предназначена для продажи акций;
d) Функция ***update()*** корректирует стоимость одной акции и пересчитывает общую стоимость акций;
e) Функция ***show()*** отображает данные о пакете акций.
Функции ***buy()*** и ***sell()*** должны обеспечивать проверку, что число проданных или купленных акций не будет отрицательным числом. Если же пользователь пытается продать акций больше, чем у него имеется, функция ***sell()*** прекращает выполнение операции.
3. Создать в функции ***main()*** несколько объектов класса ***Stock.*** Написать программу для демонстрации работы с данным классом.

Использование конструктора и деструктора

Конструктор – *это функция- элемент класса специального назначения, которая вызывается всякий раз, когда создается некоторый объект этого класса. Конструктор какого-либо класса имеет тоже имя, что и его класс. Как правило, конструктор используется для инициализации элементов объекта класса.*

Конструктор, заданный по умолчанию, *не имеет аргументов, он используется в тех случаях, когда объект создается без явной инициализации.*
//прототип конструктора по умолчанию :
myclass ();
Эта программа использует конструктор, заданный по умолчанию, для неинициализированных объектов:
myclass ob;

Конструктор с параметрами. *Конструктору можно передавать аргументы. Для этого просто добавьте необходимые параметры в объявление и определение конструктора. Затем при объявлении объекта задайте параметры в качестве аргументов.*
//прототип конструктора

student (char * fam, int age);

Его нужно использовать для инициализации новых объектов следующим образом:
//первичная форма
student st = st ("Petrov", 20);

//укороченная форма
student st("Petrov", 20);

//динамический объект
student *std= new ("Petrov", 20);

Деструктор – это функция, которая вызывается при уничтожении объекта. *Каждый класс может иметь только один деструктор. Он не имеет возвращаемого типа, не может даже иметь тип void; у него нет аргументов, а его имя является именем соответствующего класса, которому предшествует тильда.*
Например, деструктор класса myclass имеет следующий прототип:
~myclass ();

Пример:
```
#include <iostream>
using namespace std;

class myclass
{
int a;
public:
myclass (); // конструктор
myclass (int n);// конструктор с параметрами
~myclass (); // деструктор
void show() ;
}; // обратите внимание на точку с запятой в конце объявления класса!!

myclass : : myclass ( )
{
cout << "Содержимое конструктора\n";
a = 10;
}
myclass : : myclass(int n)
{
cout << "Содержимое конструктора с параметрами\n";
a = n;
}
myclass::~myclass ( )
{
cout <<"Удаление. . \n";
```

```
}
void myclass :: show()
{
cout << a << "\n";
}
int main()
{
myclass ob;// вызов конструктора
myclass ob1(44);// вызов конструктора с параметрами
ob.show();
ob1.show();
return 0;
}
```

Задание 2. ***Использование конструкторов и деструкторов***

Доработать класс ***Stock*** (см. задание 1), определив конструктор по умолчанию, конструктор с параметрами и деструктор.
Внести в конструкторы и деструктор выдачу сообщений на экран о том, какая функция была вызвана.
Выяснить время вызовов конструкторов и деструкторов.
Продемонстрировать работу программы.

Массив объектов
В случае, когда возникает необходимость иметь несколько объектов одного и того же класса, то целесообразно будет создавать массивы объектов.

myclass ob[4]; //создается массив из четырех объектов класса myclass.

Каждый элемент ***ob[0], ob[1]*** *и т.д. – это объект класса* ***myclass,*** *и поэтому, он может использоваться с методами класса* ***myclass.***

ob[4].show(); // применить функцию show() к четвертому элементу

Для ***инициализации*** *элементов массива может быть использован* ***конструктор.***

```
const int STUD=3;
student std [STUD]=
        {
         student (“Petrov”,19),
        student (“Sidorov”,20),
        student (“Nosov”,18)
        };
```

Задание 3. ***Использование массива объектов***

Доработайте класс Stock (см. Задание 1). Создайте массив из 5 объектов данного класса. Проинициализируйте элементы массива с помощью конструктора. Продемонстрируйте работу программы.

Задание 4. **Разработать классы для описанных ниже объектов в соответствии с вариантом. Включить в класс методы set (…), get (…), show (…). Определить другие методы.**

1. **Student**: Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс. Создать массив объектов. Вывести:
а) список студентов заданного факультета;
б) списки студентов для каждого факультета и курса;
в) список студентов, родившихся после заданного года.

2. **Abiturient**: Фамилия, Имя, Отчество, Адрес, Оценки. Создать массив объектов. Вывести:
а) список абитуриентов, имеющих неудовлетворительные оценки;
б) список абитуриентов, сумма баллов у которых не меньше заданной;
в) выбрать N абитуриентов, имеющих самую высокую сумму баллов, и список абитуриентов, имеющих полупроходной балл.

3. **Aeroflot**: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Создать массив объектов. Вывести:
а) список рейсов для заданного пункта назначения;
б) список рейсов для заданного дня недели;
в) список рейсов для заданного дня недели, время вылета для которых больше заданного.

4. **Book**: Автор, Название, Издательство, Год, Количество страниц. Создать массив объектов. Вывести:
а) список книг заданного автора;
б) список книг, выпущенных заданным издательством;
в) список книг, выпущенных после заданного года.

5. **Worker**: Фамилия и инициалы, Должность, Год поступления на работу, Зарплата. Создать массив объектов. Вывести:
а) список работников, стаж работы которых на данном предприятии превышает заданное число лет;
б) список работников, зарплата которых больше заданной;
в) список работников, занимающих заданную должность.

6. **Train**: Пункт назначения, Номер поезда, Время отправления, Число общих мест, Купейных, Плацкартных. Создать массив объектов. Вывести:
а) список поездов, следующих до заданного пункта назначения;
б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;
в) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

7. **Product**: Наименование, Производитель, Цена, Срок хранения, Количество. Создать массив объектов. Вывести:
а) список товаров для заданного наименования;
б) список товаров для заданного наименования, цена которых не превышает указанной;
в) список товаров, срок хранения которых больше заданного.

8. **Bus**: Фамилия и инициалы водителя, Номер автобуса, Номер маршрута, Марка, Год начала эксплуатации, Пробег. Создать массив объектов. Вывести:
а) список автобусов для заданного номера маршрута;
б) список автобусов, которые эксплуатируются больше 10 лет;
в) список автобусов, пробег у которых больше 10 000 км.