

# Лабораторная работа 1

## Создание простого класса. Работа с конструкторами и деструкторами

**Цель.** Освоение основ работы с классами. Изучение понятий «конструктор и деструктор», тестирование различных типов конструкторов, порядка их работы.

### 1. Общие указания

#### Определение класса

Класс можно определить с помощью конструкции

```
тип_класса имя_класса { компоненты класса };
```

Точка с запятой в конце ставится обязательно. В этом определении:

Имя класса является по умолчанию именем *типа объектов*. Данные или память класса определяют *состояние объекта*. Функция, являющаяся компонентом класса, называется *методом класса*. Методами класса определяются операции над объектами класса.

#### Определение методов класса

Методы класса имеют неограниченный доступ ко всем элементам класса независимо ни от порядка объявления элементов класса, ни от спецификаторов доступа. Методы могут определяться как внутри класса, так и вне его. Определение метода внутри класса не отличается от определения обычной функции. Метод, определенный внутри класса, считается по умолчанию встроенной функцией (*inline*).

Если метод определяется вне класса, то принадлежность метода классу указывается префиксом-именем класса (namespace). В классе присутствует только прототип. Метод, определенный вне класса, по умолчанию не считается *inline*-функцией. Методы могут быть перегружены и могут принимать аргументы по умолчанию. Аргументом и/или возвращаемым значением метода может быть объект того же класса, и такие объекты разрешено объявлять внутри метода.

Создадим класс *CCurrency*, который будем использовать для хранения и обработки денежных сумм.

### 2. Пример

```
// Файл CCurr.hpp (заголовок класса)
class CCurrency {
private:
    long Dollars;
    int Cents;
public:
    CCurrency();
    CCurrency (long Dol, int Cen);
    void GetAmount (long *PDol, int *PCen);
    void PrintAmount ();
    void SetAmount (long Dol, int Cen);
    CCurrency Sum(CCurrency aCurr); // Сложение 2-х денежных сумм,
                                   //то есть 2-х объектов нашего класса
};
```

```
// Файл CCurr.cpp (реализация класса)
#include <iostream>
```

```
#include <stdio.h>
#include "CCurr.hpp"

// конструктор по умолчанию
CCurrency ::CCurrency() {
    Dollars = 0;
    Cents = 0;
}

// конструктор
CCurrency ::CCurrency (long Dol, int Cen) {
    SetAmount (Dol, Cen);
}

// получить величину денежной суммы (в переменные PDol и PCen)
void CCurrency ::GetAmount (long *PDol, int *PCen) {
    *PDol = Dollars;
    *PCen = Cents;
}

// вывод денежной суммы на экран
void CCurrency ::PrintAmount () {
    cout << '$' << Dollars << '.';
    cout << Cents << '\n';
}

// назначить величину денежной суммы
void CCurrency ::SetAmount (long Dol, int Cen) {
    Dollars = Dol + Cen/100;
    Cents = Cen%100;
}

// сложить две денежные суммы
CCurrency CCurrency::Sum(CCurrency aCurr) {
    CCurrency Temp;
    Temp.SetAmount(Dollars + aCurr.Dollars, Cents + aCurr.Cents);
    return Temp;
}
```

```
// Файл main_CCurr.cpp (использование класса)
#include <stdio.h>
#include "CCurr.hpp"
int main ()
{
    CCurrency Heating(40,50);
    CCurrency Water(20,30);
    CCurrency Electricity(25,67);
    CCurrency Total;

    Total = Heating + Water + Electricity;
    Total.PrintAmount();
    return 0;
}
```

### 3. Варианты задания

Создать класс согласно варианту. Реализовать конструкторы и деструкторы, написать функцию для (сложения/ вычитания двух объектов класса см. вариант). В программе main создать несколько объектов класса с помощью разных конструкторов. Реализовать вывод объектов на печать.

№	Задание
1.	Рациональная (несократимая) дробь представляется парой целых чисел $(a, b)$ , где $a$ — числитель, $b$ — знаменатель. Создать класс <b>Rational</b> для работы с рациональными дробями. Обязательно должны быть реализованы операции: - сложения $\text{add}, (a, b) + (c, d) = (ad + be, bd)$ ; вычитания $\text{sub}, (a, b) - (c, d) = (ad - be, bd)$ ; умножения $\text{mul}, (a, b) \times (c, d) = (ac, bd)$ ; деления $\text{div}, (a, b) / (c, d) = (ad, be)$ ; сравнения $\text{equal}, \text{greater}, \text{less}$ .
2.	Создать класс <b>Date</b> для работы с датами в формате «год.месяц.день». Класс должен включать не менее трех функций инициализации: числами, строкой вида «год.месяц.день» (например, <2004.08.31 >) и датой. Обязательными операциями являются: вычитание заданного количества дней из даты, определение високосности года, присвоение и получение отдельных частей (год, месяц, день), сравнение дат (равно, до, после), вычисление количества дней между датами.
3.	Создать класс <b>Time</b> для работы со временем в формате «час:минута:секунда». Класс должен включать в себя не менее четырех функций инициализации (конструкторов): числами, строкой (например, «23:59:59»), секундами и временем. Обязательными операциями являются: вычисление разницы между двумя моментами времени в секундах, сложение времени и заданного количества секунд, вычитание из времени заданного количества секунд, сравнение моментов времени.
4.	Создайте класс <b>planet</b> , который содержит информацию о планете солнечной системы, включающий следующие атрибуты: название, диаметр, массу и расстояние от Солнца в тысячах километров в закрытой части класса. Включите в класс открытую функцию, которая возвращает расстояние от Солнца в милях и открытую функцию, которая выводит всю информацию о планете на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
5.	Создайте класс <b>file</b> , описывающий файл на диске. Закрытыми элементами класса будут: имя файла, размер в битах, атрибуты файла, дата и время создания. Включите в класс открытую функцию, которая вычисляет размер файла в байтах, килобайтах и мегабайтах, а также функцию вывода всех данных о файле на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
6.	Создайте класс <b>card</b> , который поддерживает каталог библиотечных карточек. Этот класс содержит название книги, имя автора, выданное на руки число экземпляров, дату выдачи книги читателю и дату возврата книги в библиотеку, в закрытой части класса. Включите в класс открытые функции подсчета количества дней, которые книга находится на руках читателя, и вывода всех данных о книге на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
7.	Создайте класс <b>nomenclature</b> , описывающий товары на складе магазина. Закрытыми элементами класса будут: название товара, оптовая цена, розничная наценка и количество товаров на складе. Включите в класс открытые функции подсчета возможного чистого дохода при продаже этого товара и вывода всех данных о товаре на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
8.	Создайте класс <b>persona</b> , который содержит информацию о ФИО человека, дате рождения, поле и адресе человека закрытой части класса. Включите в класс открытые функции подсчета количества дней оставшихся до следующего дня рождения и вывода данных о человеке на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
9.	Создайте класс <b>soft</b> , который содержит информацию об установленном программном обеспечении. Закрытыми элементами класса будут: название программы, разработчик, занимаемый объем, дата завершения лицензии. Включите в класс открытые функции подсчета количества дней до завершения лицензии и вывода всех данных об установленном программном обеспечении на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
10.	Создайте класс <b>car</b> , содержащий информацию об автомобилях. Закрытыми элементами класса будут: название модели, изготовитель, количество лошадиных сил, расход топлива на 100 км и массу автомобиля. Включите в класс открытые функции подсчета количества топлива необходимого для того чтобы проехать расстояние в 1000 км и вывода всех данных об автомобиле на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.

№	Задание
11.	Создайте класс <b>worker</b> , содержащий информацию о сотрудниках предприятия. Закрытыми элементами класса будут: ФИО сотрудника, табельный номер, название отдела, должность и дата принятия. Включите в класс открытые функции расчета стажа сотрудника (количество лет. Месяцев и дней) и вывода всех данных о сотруднике на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
12.	Создайте класс <b>fluid</b> , содержащий информацию о жидких веществах. Закрытыми элементами класса будут: название вещества, цвет, запах, плотность жидкости. Включите в класс открытые функции расчета массы жидкости в одном кубическом метре и вывода всех данных о жидкости на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
13.	Создайте класс <b>country</b> , который содержит информацию об имени, форме правления, численности населения и площади страны в закрытой части класса. Включите в класс открытые функции расчета плотности населения страны и вывода всех данных о стране на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
14.	Создайте класс <b>planet</b> , который содержит информацию о планете солнечной систем, включающий следующие атрибуты: название, диаметр, массу и расстояние от Солнца в тысячах километров в закрытой части класса. Включите в класс открытую функцию, которая возвращает расстояние от Солнца в милях и открытую функцию, которая выводит всю информацию о планете на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
15.	Реализовать класс <b>Cursor</b> . Полями являются координаты курсора по горизонтали и вертикали — целые положительные числа, вид курсора — горизонтальный или вертикальный, размер курсора - целое от 1 до 15. Реализовать методы изменения координат курсора, изменения вида курсора, изменения размера курсора, метод гашения и восстановления курсора.
16.	Реализовать класс <b>Rectangle</b> . Полями являются его стороны — положительные числа с плавающей точкой. Требуется реализовать операции: получения и изменения полей данных, вычисления площади, вычисления периметра, а также масштабирования четырехугольника путем умножения сторон на заданное число.
17.	Создать класс компьютерный файл <b>File</b> . Поля: дата создания, дата последней модификации, родительский каталог, объем файла, атрибуты (скрытый/системный), (полный доступ/только для чтения). Требуется реализовать операции: Создание нового файла, вычисление его "возраста", изменение атрибутов.
18.	Создайте класс для имитации секундомера. Члены класса: переменные типа double для хранения начального и конечного значения времени. Обязательны функции-члены класса для инициализации секундомера (присваивает переменным нулевые значения), для запуска и остановки секундомера, для вывода на экран величины истекшего времени.
19.	Создать класс <b>Triangle</b> для представления треугольника. Поля данных должны включать углы и стороны. Требуется реализовать операции: получения и изменения полей данных, вычисления площади, вычисления периметра, вычисления высот, а также определения вида треугольника (равносторонний, равнобедренный или прямоугольный).
20.	Создать класс <b>Point</b> для работы с точками на плоскости. Координаты точки — декартовы. Обязательно должны быть реализованы: перемещение точки по оси X, перемещение по оси Y, определение расстояния до начала координат, расстояния между двумя точками, преобразование в полярные координаты, сравнение на совпадение и несовпадение.
21.	Создайте класс, который поддерживает каталог библиотечных карточек. Этот класс должен хранить заглавие книги имя автора и выданное на руки число экземпляров книги. Заглавие и имя автора храните в виде строки символов, а количество экземпляров в виде целого числа. Включите в класс открытую функцию для запоминания информации о книгах, а также открытую функцию, которая выводит информацию на экран.
22.	Создать класс <b>Angle</b> для работы с углами на плоскости, задаваемыми величиной в градусах и минутах. Обязательно должны быть реализованы: перевод в радианы, приведение к диапазону 0-360, увеличение и уменьшение угла на заданную величину, получение синуса, сравнение углов.

## **4. Требования к отчету**

1. Обоснование выбора структуры памяти класса и его интерфейса.
2. Блок-схемы алгоритмов функций обработки.
3. Описание класса + схемы.
4. Результаты тестирования класса.

## **5. Контрольные вопросы**

1. Что значит в ООП понятие «класс», и какой формат его объявления в программе?
2. Что такое объект класса, что он содержит?
3. Какие существуют уровни доступа к объектам и методам класса (дать характеристику каждому)?
4. Перечислить основные концепции ООП, дать пояснение.