

## **Лабораторная работа №2**

### **Агрегация. Конструкторы. Перегрузка.**

*Цель лабораторной работы:* изучение структуры, свойств и видов объектов; изучение способов доступа к полям и правил вызова методов объектов; получение навыков объектно-ориентированного программирования на языке C++.

*Задание на программирование:* используя технологию объектно-ориентированного программирования разработать программу.

#### **Порядок выполнения работы:**

1) Выбрать вариант по номеру в списке группы (циклически, если группа больше 27 человек)

2) Выполнить индивидуальное задание.

3) Оформить отчет о лабораторной работе в составе:

- титульный лист;
- основная часть.

4) Основная часть должна содержать:

- номер варианта и индивидуальное задание;
- описание выбранной модели решения, сопоставление и объяснение основных частей алгоритма и выбранных для их реализации синтаксических конструкций языка, в том числе иерархию и структуру классов, методы обработки полей;

- исходный код;
- снимки экрана с результатом работы программы;

5) Обязательные элементы кода:

• Для демонстрации требуется написать главную функцию. Обязательно должны быть продемонстрированы различные способы создания объектов базового класса (все виды конструкторов). Программа должна демонстрировать использование всех методов.

• Помимо указанных в индивидуальном задании операций, в базовом классе должны быть реализованы общие методы:

– инициализации;

– ввод с клавиатуры (перегрузка инициализации, вызванный без аргументов метод инициализации, запрашивает данные от пользователя);

– вывод на экран состояния всех атрибутов класса.

Пояснение: в main должны отсутствовать операторы потокового ввода и вывода.

• Создать базовый класс с закрытыми полями, где операции реализуются как методы класса.

- Все атрибуты базового класса инкапсулировать в отдельном классе. В базовом классе должно быть только одно поле данных, представленное объектом класса атрибутов (агрегация).

- Класс атрибутов снабдить конструктором инициализации, перегрузить для него операции ввода/вывода как внешние дружественные функции.

- В базовом классе реализовать конструктор со списком инициализации и значениями по умолчанию (один или несколько) и конструктор без аргументов (вызов метода инициализации).

- Указанные в индивидуальном задании операции реализовать посредством перегрузки соответствующих операторов. В первую очередь реализуются как методы класса. Если через метод реализовать невозможно, то как внешние дружественные функции (требуется объяснить невозможность).

- Интерфейс класса и реализация должны быть разделены и вынесены в отдельные файлы, за пределы файла с главной функцией.

б) Загрузить отчёт в личный кабинет в формате PDF.

## Варианты индивидуальных заданий

1. Комплексное число представляется парой действительных чисел ( $a$ ,  $b$ ), где  $a$  — действительная часть,  $b$  — мнимая часть. Реализовать класс для работы с комплексными числами, содержащий операции:

- сложения  $(a, b) + (c, d) = (a + b, c + d)$ ;
- умножения  $(a, b) \times (c, d) = (ac - bd, ad + bc)$ ;

2. Создать класс, задаваемый тройкой координат, содержащий операции:

- сложение векторов;
- скалярное произведение векторов.

3. Создать класс для работы с моделями экранных окон. В качестве полей задаются: заголовок окна, координаты левого верхнего угла, размер по горизонтали, размер по вертикали, цвет окна. Координаты и размеры указываются в целых числах. Реализовать операции:

- передвижение окна;
- изменение цвета.

4. Создать класс для работы с денежными суммами. Число должно быть представлено двумя полями: типа `long` для рублей и типа `unsigned char` — для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать:

- сложение;
- сравнение.

5. Реализовать класс имитирующий курсор. Полями являются координаты курсора по горизонтали и вертикали — целые положительные числа, вид курсора — горизонтальный или вертикальный. Реализовать методы:

- изменения координат курсора;
- изменения вида курсора.

6. Создать класс для работы с углами на плоскости, задаваемыми величиной в градусах и минутах. Реализовать методы:

- перевод в радианы;
- увеличение и уменьшение угла на заданную величину.

7. Создать класс для работы с точками на плоскости. Координаты точки — декартовы. Реализовать методы:

- перемещение точки;
- определение расстояния до начала координат.

8. Рациональная (несократимая) дробь представляется парой целых чисел ( $a$ ,  $b$ ), где  $a$  — числитель,  $b$  — знаменатель. Создать класс для работы с рациональными дробями. Реализовать методы:

- сложения  $(a, b) + (c, d) = (ad + bc, bd)$ ;
- умножения  $(a, b) \times (c, d) = (ac, bd)$ .

9. Создать класс для работы с датами в формате «год.месяц.день». Дата представляется тремя полями типа `unsigned int`: для года, месяца и дня. Реализовать методы:

- вычисление даты через заданное количество дней;

– присвоение и получение отдельных частей (год, месяц, день).

10. Создать класс для работы со временем в формате «час:минута:секунда». Реализовать методы:

- сложение времени и заданного количества секунд;
- перевод в минуты (с округлением до целой минуты).

11. Реализовать класс для работы с нечеткими числами, которые представляются тройками чисел  $(x - e_1, x, x - e_2)$ . Для чисел  $A = (A - a_1, A, A - a_2)$  и  $B = (B - b_1, B, B + b_2)$  реализовать методы:

- сложение  $A + B = (A + B - a_1 - b_1, A + B, A + B + a_2 + b_2)$ ;
- умножение  $A \cdot B = (A \times B - B \times a_1 - A \times b_1 + a_1 \times b_1, A \times B, A \times B + B \times a_1 + A \times b_1 + a_1 \times b_1)$ .

12. Реализовать класс, представляющий собой банковский счет. В классе должны быть четыре поля: фамилия владельца, номер счета, процент начисления и сумма в рублях. Необходимо выполнять следующие операции:

- сменить владельца счета;
- снять некоторую сумму денег со счета;
- положить деньги на счет.

13. Номиналы российских рублей могут принимать значения 1, 2, 5, 10, 50, 100, 500, 1000, 5000. Копейки представить как 0.01 (1 копейка), 0.05 (5 копеек), 0.1 (10 копеек), 0.5 (50 копеек). Создать класс для работы с денежными суммами. Сумма должна быть представлена полями-номиналами, значениями которых должно быть количество купюр данного достоинства. Дробная часть (копейки) при выводе на экран должны быть отделена от целой части запятой. Реализовать:

- сложение сумм;
- умножение на число;

14. Реализовать класс, моделирующий работу банкомата. В классе должны содержаться поля для хранения идентификационного номера банкомата, информации о текущей сумме денег, оставшейся в банкомате. Сумма денег представляется полями-номиналами 10-1000 (см. задание 13). Реализовать методы:

- загрузки купюр;
- снятия определенной суммы денег. Метод снятия денег должен выполнять проверку на корректность снимаемой суммы.

15. Создать класс для работы с дробными числами. Число должно быть представлено двумя полями: целая часть — длинное целое со знаком, дробная часть — беззнаковое короткое целое. Реализовать методы:

- вычитания;
- деления.

16. Создать класс имитацию товарооборота. В классе должны быть представлены поля: наименование товара, цена товара, количество единиц товара. Реализовать методы:

- увеличения и уменьшения товара;
- вычисления стоимости товара.

17. Создать класс для работы с 64-битовыми строками. Битовая строка должна быть представлена двумя полями типа `unsigned long`. Реализовать операции для работы с битами:

- `and`;
- сдвиг влево на заданное количество битов.

18. Создать класс имитацию подсчёта выплат. В классе должны быть представлены поля: фамилия-имя-отчество, год поступления на работу, оклад, количество отработанных дней в месяце, количество рабочих дней в месяце. Реализовать методы:

- вычисления начисленной суммы от отработанных дней;
- вычисления стажа (полное количество лет, прошедших от года поступления на работу, до текущего года);
- начисления надбавки.

19. Комплексное число представляется парой действительных чисел  $(a, b)$ , где  $a$  – действительная часть,  $b$  – мнимая часть. Реализовать класс для работы с комплексными числами, содержащий операции:

- вычитания  $(a, b) - (c, d) = (a - b, c - d)$ ;
- деления  $(a, b) / (c, d) = (ac + bd, bc - ad) / (c^2 + d^2)$ ;

20. Создать класс, задаваемый тройкой координат, содержащий операции:

- вычитание векторов;
- умножение на скаляр.

21. Создать класс для работы с денежными суммами. Число должно быть представлено двумя полями: типа `long` для рублей и типа `unsigned char` – для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать:

- вычитание
- деление сумм.

22. Рациональная (несократимая) дробь представляется парой целых чисел  $(a, b)$ , где  $a$  — числитель,  $b$  — знаменатель. Создать класс для работы с рациональными дробями. Реализовать методы:

- вычитания  $(a, b) - (c, d) = (ad - bc, bd)$ ;
- деления  $(a, b) / (c, d) = (ad, bc)$ .

23. Реализовать класс для работы с нечеткими числами, которые представляются тройками чисел  $(x - e_1, x, x + e_2)$ . Для чисел  $A = (A - a_1, A, A + a_2)$  и  $B = (B - b_1, B, B + b_2)$  реализовать методы:

- вычитание  $A - B = (A - B - a_1 - b_1, A - B, A - B + a_2 + b_2)$ ;
- деление  $A / B = ((A - a_1) / (B + b_2), A / B, (A + a_2) / (B - b_2))$ .

24. Номиналы российских рублей могут принимать значения 1, 2, 5, 10, 50, 100, 500, 1000, 5000. Копейки представить как 0.01 (1 копейка), 0.05 (5 копеек), 0.1 (10 копеек), 0.5 (50 копеек). Создать класс для работы с денежными суммами. Сумма должна быть представлена полями-номиналами, значениями которых должно быть количество купюр данного достоинства. Дробная часть (копейки) при выводе на экран должны быть отделена от целой части запятой. Реализовать:

- вычитание сумм;
- деление суммы на число.

25. Создать класс для работы с дробными числами. Число должно быть представлено двумя полями: целая часть — длинное целое со знаком, дробная часть — беззнаковое короткое целое. Реализовать методы:

- сложения;
- умножения.

26. Создать класс для работы с 64-битовыми строками. Битовая строка должна быть представлена двумя полями типа `unsigned long`. Реализовать операции для работы с битами:

- ог;
- сдвиг вправо на заданное количество битов.

27. Создать класс имитацию расчёта налоговых вычетов. В классе должны быть представлены поля: фамилия-имя-отчество, начисленная сумма, два поля налогов. Первый налог берётся от начисленной суммы. Второй от оставшейся суммы. Реализовать методы:

- вычисления суммы, выдаваемой на руки;
- субсидии (увеличение суммы выдаваемой на руки, облагается только налогом №2)