

## Язык программирования Си++

Иванов А.П., Князева О.С.

### Семинар 4. Коллоквиум: методика его проведения и типовые вопросы.

Коллоквиум проводится на семинарском занятии в середине семестра (обычно – до 1-го апреля) в форме тотального опроса с билетами. На семинаре, следующим за коллоквиумом, рекомендуется провести разбор типичных ошибок студентов при ответе на вопросы.

#### 1. Билеты

---

##### 1. Билет

- 1) Объектно–ориентированное программирование: инкапсуляция, наследование, полиморфизм.
- 2) Динамическая память. Операторы `new` и `delete`.

---

##### 2. Билет

- 1) Классы. Конструкторы, деструктор.
- 2) Передача и возврат параметров в функции по значению, по ссылке, по указателю.

---

##### 3. Билет

- 1) Статические методы и поля.
- 2) Наследование, модификаторы доступа `private`, `protected`, `public`.

---

##### 4. Билет

- 1) `Explicit`-конструкторы.
- 2) Абстрактные классы.

---

##### 5. Билет

- 1) Оператор «`:`». Определение тела методов вне класса.
- 2) Потоки ввода–вывода Си++ `cin` и `cout` и их использование.

---

##### 6. Билет

- 1) Перегрузка операторов и функций.
  - 2) Ссылки.
- 

---

##### 7. Билет

- 1) Перегрузка операторов «`+`» и «`+=`», ключевое слово `this`.
- 2) Передача и возврат параметров в функции по значению, по ссылке, по указателю.

---

##### 8. Билет

- 1) Динамическая память. Операторы `new` и `delete`.
- 2) Дружественные классы и функции (`friend`).

---

##### 9. Билет

- 1) Ссылки.
- 2) Полиморфизм, виртуальные функции.

---

##### 10. Билет

- 1) Передача и возврат параметров в функции по значению, по ссылке, по указателю.
- 2) Множественное наследование.

---

##### 11. Билет

- 1) Перегрузка оператора приведения типа.
- 2) Ссылки.

---

##### 12. Билет

- 1) Указатели.
  - 2) Полиморфизм, виртуальные функции.
-

---

**13. Билет**

- 1) Наследование, модификаторы доступа `private`, `protected`, `public`.
  - 2) Перегрузка операторов ввода–вывода.
- 

**14. Билет**

- 1) Дружественные классы и функции (`friend`).
  - 2) Виртуальный деструктор.
- 

**15. Билет**

- 1) Полиморфизм, виртуальные функции.
  - 2) Динамическая память. Операторы `new` и `delete`.
- 

**16. Билет**

- 1) Виртуальный деструктор.
  - 2) Передача и возврат параметров в функции по значению, по ссылке, по указателю.
- 

**17. Билет**

- 1) Значения аргументов функций по умолчанию.
  - 2) Виртуальный деструктор.
- 

**18. Билет**

- 1) Чисто виртуальные функции, абстрактные классы.
  - 2) Указатели.
- 

**19. Билет**

- 1) Множественное наследование.
  - 2) Дружественные классы и функции (`friend`).
- 

---

**20. Билет**

- 1) Виртуальное наследование.
  - 2) Оператор «`::`». Определение тела методов вне класса.
- 

**21. Билет**

- 1) Перегрузка операторов ввода–вывода.
  - 2) Динамическая память. Операторы `new` и `delete`.
- 

**22. Билет**

- 1) Перегрузка операторов «`+`» и «`+=`», ключевое слово `this`.
  - 2) Чисто виртуальные функции, абстрактные классы.
- 

**23. Билет**

- 1) Значения аргументов функций по умолчанию.
  - 2) Виртуальное наследование.
- 

**24. Билет**

- 1) `explicit`-конструкторы.
  - 2) Множественное наследование.
- 

**25. Билет**

- 1) Перегрузка оператора приведения типа.
  - 2) Наследование, модификаторы доступа `private`, `protected`, `public`.
-

## 2. Факультативные задания

Так как коллоквиум проводится на семинарском занятии, то у студентов появляются дополнительные 2 недели, которые следует использовать для доделки и сдачи всех выданных ранее задач. Практика показывает, что к этому моменту «хвосты» имеет, как минимум, половина группы.

Студенты, которые успешно сдали все предыдущие задания, могут получить факультативное задание по теме «полиморфизм». Учет этих факультативных задач отличается от учета обычных заданий: они не являются обязательными, то есть, их отсутствие – не ухудшает оценки за успеваемость в течение семестра, хотя их наличие – улучшает эту оценку. Если студент не успевает их сдать до 5-го семинара, то такому студенту рекомендуется сначала сдавать плановые, обычные задания и лишь когда плановые задания сданы – возвращаться к выполнению факультативных заданий.

---

### 1. Вариант

Создайте абстрактный класс – числа. Определите в нем чистую виртуальную функцию – модуль числа. Создайте на его базе 2 производных класса – рациональные числа и комплексные числа. Создайте массив, которые содержит указатели на объекты этих классов. Напишите функцию вычисляющую сумму модулей чисел в массиве.

---

### 2. Вариант

Создайте абстрактный класс – уравнения. Определите в нем чистую виртуальную функцию `solve()` – решение уравнения. Создайте на его базе 2 производных класса – линейные уравнения вида  $ax+b=0$  и уравнения вида  $\sin(x)=a$ . Создайте массив, которые содержит указатели на объекты этих классов. Напишите функцию вычисляющую интервал, в котором лежат решения уравнений в массиве.

---

### 3. Вариант

Создайте абстрактный класс – геометрических объектов. Определите в нем чистую виртуальную функцию – площадь фигуры. Создайте на его базе 2 производных класса – окружность на плоскости и ромб на плоскости. Создайте массив, которые содержит указатели на объекты этих классов. Напишите функцию, вычисляющую суммарную площадь всех объектов в массиве.

---

### 4. Вариант

Создайте абстрактный класс – геометрических объектов. Определите в нем чистую виртуальную функцию – объем объекта. Создайте на его базе 2 производных класса – куб и сфера. Создайте массив, которые содержит указатели на объекты этих классов. Напишите функцию, вычисляющую суммарный объем всех объектов в массиве.

---

### 5. Вариант

Создайте абстрактный класс – геометрических объектов. Определите в нем чистую виртуальную функцию – площадь фигуры. Создайте на его базе 2 производных класса – равносторонний треугольник и квадрат. Создайте массив, которые содержит указатели на объекты этих классов. Напишите функцию, вычисляющую суммарную площадь всех объектов в массиве.

---

### 6. Вариант

Создайте абстрактный класс – геометрических объектов. Определите в нем чистую виртуальную функцию – объем объекта. Создайте на его базе 2 производных класса –

пирамида и цилиндр. Создайте массив, которые содержит указатели на объекты этих классов. Напишите функцию, вычисляющую суммарный объем всех объектов в массиве.

---

### **7. Вариант**

Создайте абстрактный класс – геометрических объектов. Определите в нем чистую виртуальную функцию – поворот фигуры на указанное количество градусов. Создайте на его базе 2 производных класса – равносторонний треугольник и квадрат. Создайте массив, которые содержит указатели на объекты этих классов. Напишите функцию, поворачивающую каждый объект массива на один и тот же угол.

---

### **8. Вариант**

Есть два набора «зверей» – волки и зайцы. Количество зверей определяется пользователем в самом начале (например, параметрами командной строки).

Рождаются все существа в случайном месте экрана 25 рядов на 80 столбцов, за один ход волк может переместиться на 2 клеточки в случайном направлении, заяц перемещается на 1 клетку в случайном направлении (направление – одна из четырех сторон света).

Граничные условия – периодические, т.е. если существо переместилось за границу экрана, то оно появляется на противоположной его стороне.

Заяц живет 10 ходов. Волки живут 20 ходов. Волк, съевший 5 зайцев размножается, порождая еще одного волка в той точке, где он находится. Зайцы размножаются каждые 5 ходов – в тех клетках, где они находятся. Заяц может быть съеден волком, если он попал на одну клетку с волком, если волков и зайцев в клетке несколько, то первый же волк съедает всех зайцев.

Необходимо создать абстрактный класс – «зверь» и 2 производных класса – «волки» и «зайцы».

Все звери хранятся в массиве указателей на их абстрактный базовый класс, реализующий все нужные методы (в базовом классе соответствующие методы должны быть чистыми) и содержащий все данные о возрасте, сытости и положении существа на экране.

Считаем, что между кадрами проходят одинаковые интервалы времени, каждый кадр рассчитывается в виде массива символов 25x80 (заяц \*, волк #, остальное – пробел) и выводится на экран с небольшой задержкой.

Задание может варьироваться заданием разных правил взаимодействия объектов: волки могут при каких-то условиях поедать друг друга, несколько зайцев могут убить волка, находящегося не далее некоторого расстояния от них и т.п.