

ООП. С++. Вопросы для проверки и закрепления знаний.

1. Проанализируйте ранее разработанные Вами программы и ответьте на вопрос: в каких из них наблюдается слабая связь между структурой данных и функционированием алгоритма? Как следует модифицировать программы для устранения этого разрыва?

2. Проанализируйте ранее разработанные Вами программы и ответьте на вопрос: как часто Вы в своих программах используете иерархическое представление данных для образования новых типов?

Как следует изменить описание данных, чтобы это стало возможным?

3. Проанализируйте ранее разработанные Вами программы и ответьте на вопрос: какие данные Вы хотели бы скрыть от случайной модификации другими функциями? От пользователей программы?

4. Объекты в С++ часто отражают реальные физические объекты или математические понятия. Какие объекты Вы можете выделить в Ваших программах?

5. Проанализируйте ранее разработанные Вами программы и приведите пример иерархического наследования типов.

6. Приведите пример, где наследование типов носит более сложный характер, чем классификация.

7. Проанализируйте ранее разработанные Вами программы и ответьте на вопрос: какие функции удобно было бы назвать одним именем?

8. В следующей программе определите контекст, видимость и продолжительность каждой переменной:

```
static double x = 2.0;
int k = 3;
double func(double x, int k)
{
    if (k < 0) return 0.0;
    else{
        for (int i=0; i < k; x *= x, i++);
        return x;
    }
};
main(void)
{
    int k = 5;
    cout << func(x,k) << "\n";
```

```
    cout << func(x,::k) << "\n";  
}
```

9. Разработчики компилятора языка VC++ утверждают, что эффективность операций `new` и `delete` выше, чем функций `malloc()` и `free()`. Напишите программу, позволяющую сравнить быстродействие `new` и `delete` с `malloc()` и `free()`.

10. Напишите программу для определения сколько байт на Вашей системе занимают ссылочные переменные. Ответ на этот вопрос даст Вам рекомендацию по использованию ссылок вместо больших аргументов функций.

11. Проанализируйте ранее написанные Вами программы и ответьте на вопрос: в каких из них возможно использование умалчиваемых аргументов функций и значение последних.

12. В каких случаях выражение в операторе вывода `"<<"` следует брать в скобки? Подсказка: для оператора вывода `"<<"` сохраняется приоритет предопределенной операции сдвига влево.

13. Напишите программу для сравнения эффективности операторов ввода-вывода `>>` и `<<` языка C++ с аналогичными функциями `scanf()` и `printf()`.

14. Определите быстродействие на Вашем компьютере всех возможных операций для различных арифметических типов: `char`, `unsigned char`, `int`, `unsigned int`, `long`, `unsigned long`, `float`, `double`, `long double`. Проанализируйте результаты, сделайте выводы и руководствуйтесь ими в последующей работе.

15. Напишите компоненту-функцию `void Array::put(int i,int x)`; которая помещает значение `x` в `i`-й элемент массива. Какие еще компоненты-функции Вы хотели бы определить в классе `Array`? Напишите их.

16. Напишите программу, которая для класса `Array` позволяет оценить эффективность механизма встраиваемых функций как с точки зрения быстродействия, так и размера кода.

17. Перепишите определение класса `Array` для случая, когда все функции являются "друзьями" класса. Изменяются ли при этом атрибуты доступа компонент-данных?

18. Как следует модифицировать класс `Array`, чтобы можно было определить количество обращений к элементам массива? Количество вызовов функций объектов?

19. Разработайте класс `Stack` как стек целых чисел фиксированного размера, определите функции `pop()` и `push()`, напишите программу для его использования.

20. Определите для класса конструктор `String::String(int size)`, принимающий в качестве аргумента длину строки `size`.

21. Опишите класс для стека строк символов. Выполните две реализации. В первой длина строки фиксирована и размер стека передается конструктору в качестве параметра. Во второй реализации предусмотрите ситуацию, когда длина строки произвольная и заранее не известен размер стека.

22. Опишите класс для массива символьных строк. Выполните две реализации как в предыдущем примере.

23. Определите конструкторы копирования для классов из вопросов 21 и 22.

24. Укажите, какие компоненты базовых классов доступны в производном классе D?

```
class B1 {
    int a;
    float f;
protected:
    char ch;
    double d;
};
class B2 {
protected:
    float pi;
    int x;
public:
    char* str;
};
class D: public B1, private B2 {...}
```

25. Добавьте в описание графической системы класс, определяющий цвета фигур, а также классы других библиотечных графических фигур.

26. Добавьте в описание графической системы функцию высвечивания фигуры Show() и "гашения" изображения фигуры Hide(). Подсказка: "гашения" изображения фигуры на экране заключается в ее перерисовке с цветом, совпадающим с цветом фона.

27. Укажите порядок вызова конструкторов и деструкторов при создании и уничтожении объекта Arc.

28. Определите класс Array как множество целых чисел. Перегрузите для класса Array операции +, /, и == для объединения, вычитания и сравнения множеств соответственно.

29. Выполните упражнение 6.1. для функций-компонент и функций-друзей.

30. Многократная перегрузка операций является стандартным приемом в библиотечных классах, когда аргументами операций могут быть операнды разных типов. Выполните перегрузку всех арифметических операций класса `complex` для возможности работы с операндами типа `double`. Подсказка: для каждой бинарной операции следует предусмотреть варианты `(complex, complex)`, `(double, complex)`, `(complex, double)`.

31. Выполните перегрузку комбинированных операций присваивания для класса `complex`. Предусмотрите вариант, когда аргумент имеет тип `double`.

32. Выполните перегрузку операций `new` и `delete` для класса `complex` так, чтобы сразу выделялся некоторый блок связанных участков памяти. Всякое следующее обращение к `new` возвращает указатель на свободный участок блока. Операция `delete` возвращает участок в список свободных участков.

33. Выясните, будет ли на Вашей системе обнаружена ошибка, если перегрузить две функции с одинаковыми именами, но разными возвращаемыми значениями, или перегрузка будет выполнена правильно.

34. Перепишите пример графической системы из упражнения 5.1, объявив функции `Show()` и `Hide()` в классе `Point` как виртуальные.

35. Определите для Вашей системы эффективность использования виртуальных функций по сравнению с обычным вызовом функции. Сравните размер выполняемого кода и время вызова функции.

36. Напишите программу, которая любое введенное значение в не преобразованном виде выводит на экран.

37. Выполните упражнение 36, используя промежуточный буфер.

38. Напишите программу, которая во введенной строке распознает все идентификаторы языка C++ и выводит их на экран.

39. Используя потоки ввода-вывода языка C++, напишите программу для ввода анкетных данных группы людей (сотрудников, студентов и т.п.) и вывода их на экран в виде таблицы.

40. Напишите программу, которая создает все типы файлов : обычный, только для чтения, скрытый, системный и архивный. К каким из этих файлов возможен доступ из других программ ?

41. Напишите программу вывода на печать текстового файла. Предусмотрите различные возможные ситуации: не подключен принтер, отсутствует бумага, нет готовности устройства и т.д.

42. Напишите программу, которая по указанному номеру страницы текстового файла выводит ее на экран.

43. Напишите программу, которая вначале считывает пять строк, а затем, воспринимая каждую строку как поток, из первой строки считывает три целых, из второй - два плавающих, из третьей - строку символов, из четвертой - один символ, из пятой - целое, плавающее, символ и строку.

44. Напишите программу для одновременного просмотра на экране двух текстовых файлов. Предусмотрите возможность постраничного листания каждого из них.

45. Разработайте шаблоны классов стека, очереди и бинарного дерева для числовых типов.

46. Выполните предыдущее упражнение для строк символов, комплексных и двоично-десятичных чисел.