

Ответы на контрольные вопросы:

void - тип без значения
float - 4 байта, диапазон от
+/- 3.4E-38 до 3.4E+38.

Лабораторная работа ~2

Тема: Классы и объекты. Инкапсуляция.

Перегрузка операторов.

Цель: Научиться программировать на языке C++ классы, инкапсулировать данные и методы, перегружать унарные и бинарные операторы.

Контрольные вопросы:

1. Чем отличаются классы и структура в C++ технически и концептуально?

В C++ основная разница между структурами и классами - это модификатор доступа, который используется по умолчанию для их членов. Для классов, по умолчанию используется модификатор private, а

для структур - public.

Как правило, (но это не означает, что этого не может быть), структура не имеет конструкторов и методов.

Концептуальное отличие: класс обычно используется для реализации "жесткого языка" с определенными инвариантами, а структура обычно используется как "мягкий язык" с данными.

2. Чем отличаются понятия "класс" и "объект"?

Приведите примеры.

Класс - некая абстракция, логическая структура, описывающая поведение и характеристики. Напр., машина может ехать, сигнализировать и т.д.
Объект - конкретный экземпляр класса.

Например, конкретная машина.

Класс: фрукт; Объект: киви, банан, манго.

Объект

Объект - экземпляр класса
существовать из реального мира
Конкретная сущность
Объект создается
объект как правило,
можно создавать
много раз.

Класс

Класс - шаблон, по которому
создаются объекты
Группа похожих объектов
Логическая сущность
Класс описывается
Класс определяет единичный

Можно описать класс "Точка". У точки есть
координаты (x, y) . Каждая точка - это
объект класса Точка.

Другой пример, класс Сотрудник с атрибутами
(ФИО, дата рождения, зарплата). Объект класса
Сотрудник ("Иван Иван Иванович", "слесарь",
50000). Объект класса так же называется
"экземпляр класса".

3. Опишем традиционную файловую
структуру ООП проекта.

- 1) Заголовочный файл (.h) - основной
способ кодировать к программе типы
данных, структура, прототипы функций и т.д.
- 2) Файл реализации (.cpp) - файл, в
котором реализуется основная часть программы

4. Что такое header guards (include guards)?

Каково предназначение они позволяют решить, и каков есть второй способ ее решить?

Header guards (защита подключения в файле `++`) - это директива условной компиляции:

```
#ifndef SOME_UNIQUE_NAME_HERE
```

```
#define SOME_UNIQUE_NAME_HERE
```

Необходим для избежания повторного объявления. (для избежания проблем с "глобальным подключением")

Большинство компиляторов поддерживают более простую, альтернативную форму header guards - директиву `#pragma once`.

5. Что такое инкапсуляция? Какие есть спецификаторы доступа и что они означают?

В ООП инкапсуляция (сокращение информации) -

это процесс скрытого хранения деталей реализации объекта.

Инкапсуляция - механизм языка, позволяющий

связывать данные и методы, работающие с
этими данными в едином объекте.

Спецификацией доступа (изменят доступ к
данным):

- public - доступен всем;
- protected - доступен только классу и
дочерним классам;
- private - доступен только классу, которому
они принадлежат.

6. Что такое конструктор класса? Деструктор
класса? Типы конструкторов.

Конструктор класса - это особый тип метода
класса, который автоматически вызывается
при создании объекта этого же класса.

Конструкторы имеют определенные правила
их именования:

- конструктор всегда должен иметь то же
имя, что и класс (указываются верхний и
нижний регистр);
- конструктор не имеет типа возврата (data void).

Деструктор - это специальный тип метода класса, который выполняется при удалении объекта класса.

Деструкторы имеют свои правила, кот. касаются их имен:

- деструктор должен иметь то же имя, что и класс, со знаком тильды (~) в самом начале;
- деструктор не может принимать аргумента;
- деструктор не имеет типа возврата.

Типы конструкторов:

- конструктор по умолчанию - не имеют параметров (или содержит параметра, кот. все имеют значение по умолчанию);
- конструктор с параметрами;
- конструктор с копирующим конструктором;
- a) - конструктор перемещения.

7. Что означает указатель `this`?

Указатель `this` - это указатель на адрес объекта класса, при этом он является скрытым первым параметром любого метода класса (кроме статических методов), а типом указателя выступает имя класса.

8. В чем заключается особенность статических методов и функций классов?

С использованием спецификатора `static` можно задать статическую функцию-член. Такую функцию можно вызывать, используя имя класса. Ее особенностью является то, что она может сослаться только на статические переменные и функции, принадлежащие классу.

Статические методы и функции класса можно рассматривать как модальную переменную класса.

Статическую функцию-член можно вызвать без ссылки на объект класса.

9. Как производится перегрузка методов и перегрузка операторов?

Перегрузка методов - необходимость создать один и тот же метод, но с разным набором параметров.

~~public void~~

add(int x, int y);

add(int x, int y, int z);

Перегрузка операторов позволяет определить действие, кот. будет выполнять оператор.

Перегрузка операторов - можно так же

Не перегружаемые: new, is, -=, +=, <=>

>>=, !=, &=, %=, /=, ., .:, *=, sizeof, typeof.

public static Time operator+ (Time t1, Time t2).

10. Укажите особенности работы с константными объектами классов.

Объекты классов можно сделать константными (используя ключевое слово const). Инициализация выполняется через конструктор класса. Как только константный объект класса

инициализируется через конструктор, но любая попытка изменить переменные - члены объекта запрещена, т.к. это нарушает принцип константности объекта.

11. Как можно запретить создание объекта класса?

1) Сделать конструктор класса private.

~~2) Сделать абстрактным классом, создать нельзя.~~

2) Если сделать класс абстрактным (нельзя будет создавать его собственные экземпляры, только классов-наследников).

12. В чем отличие глубокого копирования от поверхностного (deep and shadow copying)?

При глубоком копировании память сначала выделяется для копирования адреса, который содержит исходный указатель, а затем для копирования фактического значения.

При поверхностном копировании указатель копируется только адрес указателя -

никаких действий по содержанию адреса
указателе не предпринимается.