

Ответы на контрольные вопросы

На чем основано ООП

ООП - методология программирования, основной единицей которой является класс.

Класс - сложная сущность, состоящая из множества функций(методов) и переменных(полей).

Аналогично структурам, можем создавать экземпляры класса(объекты).

Какие аспекты стоит учитывать при проектировании классов

Инкапсуляция

Инкапсуляция - сокрытие некоторых функций объектов класса для пользователей.

В качестве аналогии можно рассмотреть мобильный телефон. Внутри любого телефона установлен аккумулятор. Однако, пользователи не имеют возможности повысить/понижить напряжение питания аппарата. Ведь при выставлении неправильного напряжения может произойти поломка.

При проектировании класса тоже следует скрывать некоторый функционал от пользователя. Для этого следует использовать спецификаторы доступа:

- private - пользователь не имеет доступа к полям/методам, определенной в этой секции
- protected - в данном рассмотрении аналогичен private
- public - поля/методы доступны вне класса.

Не принято определять публичные поля

Поля классов должны быть расположены в private секции.

Если пользователю необходимо получать значение данного поля пишем геттер:

...

```
type getFieldName() const
{
    return field;
}
...
```

По аналогии, если пользователю нужно дать возможность изменять значения поля - пишем сеттер. Для лучшей читаемости следует выносить определение больших методов за класс

Почему удобно разделять классы на интерфейс и реализацию?

1. Улучшается читаемость кода
2. При создании библиотек можем предоставлять пользователю лишь header, и скомпилированную версию src файла. Таким образом, защищаем наш код от плагиата.
3. Легче разрешать ситуации вида: метод первого класса принимает объект второго, а метод второго - объект первого.

Чем внутреннее связывание отличается от внешнего?

Сущность обладает внешним связыванием, если она видна всем единицам трансляции.

При этом, стоит определять данную сущность в отдельной единице трансляции(не в файле с объявлением). Иначе может возникнуть ошибка многократного включения

Сущность обладает внутренним связыванием если она видна лишь в конкретной единице трансляции.

Для объявления сущностей с внешним связыванием используется ключевое слово ``static``. С внутренним - ``extern`` (зачастую использование этого слова не является необходимым - например у функций уже присутствует разделение на объявление и определение).

Какими особенностями обладают именованные пространства имён?

1. Пространство может быть объявлено в нескольких файлах. Это удобно для расширения данного пространства.
2. Пространство имён может быть вложено в другое пространство имён.
3. Ключевое слово ``using`` позволяет не использовать оператор доступа к области видимости внутри данного файла.