

# Принципы ООП Абстракция и интерфейсы. Пример проектирования



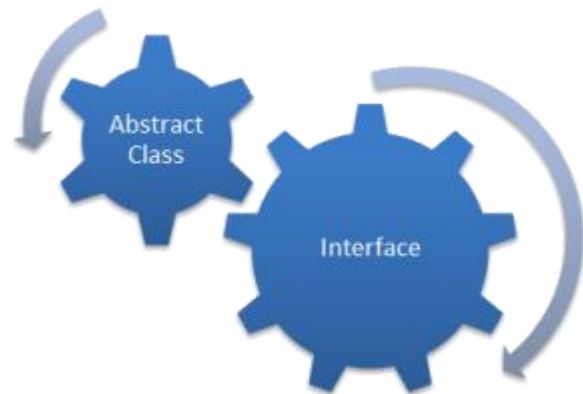
# Принципы ООП.

Абстракция и интерфейсы.



# Сегодня на занятии:

1. Поговорим о проблеме множественного наследования;
2. Разберемся в понятиях <абстрактный класс> и <интерфейс> и что и когда лучше применять;
3. Ответим на некоторые вопросы, которые могут задать на реальных собеседованиях.



# Вопросы с собеседования: Что такое сигнатура метода? Что такое перегрузка метода?

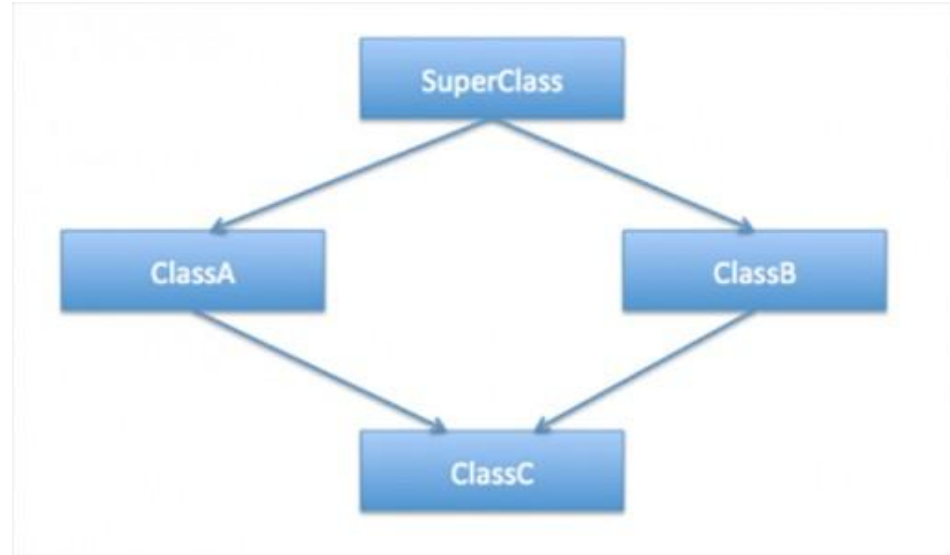


Перегрузка – создание 2 и более методов с одинаковыми именами но разными сигнатурами

# Оператор <> и проблема множественного наследования.

Классы в java могут наследоваться от:

1. Других классов;
  2. Абстрактных классов;
  3. Интерфейсов.
- Ключевые слова `extends` и `implements`.



# Может ли у абстракции быть готовая реализация?

1. Сделайте класс `Animal` из прошлого семинара абстрактным. Добавьте метод `getIll()` с простой реализацией;
2. Сделайте метод `hunt()` абстрактным. Как вы думаете, какой метод (методы) также могли бы быть абстрактными?
3. Создайте класс `VeterinaryClinic` и поле, хранящее в себе список животных - пациентов клиники. Подумайте над модификатором доступа.
4. Создайте метод `addAnimal()`, позволяющий принять в клинику нового пациента.
5. Создайте класс `Human` и унаследуйте его от `Animal`
6. Заполните список пациентов разными животными (добавьте туда и экземпляр `Human`) и удостоверьтесь, что они “больны”, вызвав метод `getIll()` у каждого. Какую логическую проблему вы тут видите?

# Человек может болеть, но не может лечиться у ветеринара.

1. Создайте интерфейс `Diseaseable` с методом `getIll()` и уберите у `Human` наследование от `Animal`.
2. Добавьте имплементацию `Diseaseable` всем животным и человеку.
3. Подумайте, как животным можно добавить эту имплементацию.
4. Создайте интерфейс `Speakable`.  
Перепроектируйте объекты - наследники `Animal`, с учётом нового интерфейса.
5. У ветеринарной клиники добавьте метод получения всех, способных говорить.



# Вопросы на собеседовании:

1. В чем различия интерфейса и абстрактного класса?
2. Когда лучше применить интерфейс, а когда абстрактный класс?
3. Все ли абстрактные методы должны быть переопределены у наследника?





# Различия абстрактного класса и интерфейса



**\*в java** Интерфейс может содержать поля, но они автоматически являются статическими (static) и неизменными (final). Все методы и переменные неявно объявляются как public.

Как Ваше  
настроение?

---





## Ваши вопросы и пожелания :)

1. Было ли сложно?
2. Получилось ли узнать/закрепить что-то новое, полезное?
3. Что не понравилось на занятии?
4. А что понравилось?
5. Что можно улучшить? Есть ли у Вас предложения?
6. Буду рад любым отзывам под записью занятия, это мотивирует! :)

**До  
встречи!**

