

ait-tr /
cohort44E

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights



main ▾

cohort44E / basic_programming / lesson_21 / theory.md



Andy179176 20240508

7 hours ago



50 lines (36 loc) · 3.44 KB

Preview

Code

Blame

Raw



🔗 Наследование 2

Абстрактные классы и абстрактные метода

Предположим у нас есть три класса: Programmer, QAEngineer, Manager. Все наши программисты, инженеры QA и менеджеры являются (IS-A) работниками и наследуют от класса Employee. Каждый работник имеет свой способ работать: программист пишет код, QA инженер проверяет, менеджер руководит и т.д. Т.е. в программе каждый из классов определяет собственный метод work(). Но, что бы метод work() был доступен у всех работников, он должен быть объявлен в классе-родителе т.е. в Employee

```
public void work(){} 
```



В такой реализации есть 2 проблемы:

1. Необходимость "пустой" реализации метода work() для Employee, ведь мы не знаем как работает неопределенный работник.
2. Мы можем создать объект Employee, такого "неопределенного работника"

Что бы избежать такой ситуации в Java придумали **абстрактные классы** и **абстрактные методы**

Класс, объявленный с указанием ключевого слова **abstract** является *абстрактным классом*, что означает, что Вы не можете создать объект такого класса.

В абстрактном классе можно объявить **абстрактный методы**, для этого необходимо в заголовке метода указать ключевое слово **abstract**. Такие методы не требуют реализации непосредственно при объявлении, они должны быть реализованы в дочерних классах.

```
public abstract class Employee {  
    public abstract void work();  
}
```



Важно

1. Несмотря на то, что объект абстрактного класса не может быть создан, в абстрактном классе может/должен быть конструктор.
2. В абстракте классе могут быть как абстрактные, так и обычные методы.
3. Абстрактный класс может не содержать абстрактных методов (подумайте, зачем тогда мы делаем класс абстрактным)

Транзитивное наследование

Если класс В наследует от класса А, а класс С наследует от класса В, то класс С также унаследует все свойства и методы класса А через цепочку наследования. Это называют **транзитивным наследованием**. Говорят: *"класс С транзитивно наследует от класса А"*

Все классы в Java транзитивно или напрямую наследуют от класса Object