

ait-tr /
cohort44E

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

cohort44E / basic_programming / lesson_20 / theory.md



Andy179176 20240507

5 hours ago



76 lines (62 loc) · 4.44 KB

Preview

Code

Blame

Raw



Наследование

Наследование - это концепция ООП, которая позволяет создавать новые классы на основе уже существующих классов. Новый класс, называемый подклассом, наследником или дочерним классом может наследовать свойства и методы существующего класса, называемого суперклассом или родителем. Подклассы могут добавлять новые методы и поля, а также переопределять методы суперкласса для реализации специфической для них функциональности.

Преимущества использования наследования:

1. Наследование позволяет объединить разные типы данных (дочерние классы) в рамках одного (родительского) типа данных. Таким образом, может быть реализован общий, единый подход к обработке всех объектов, вне зависимости к какому из подклассов принадлежит конкретный объект.
2. **Переиспользование кода.** Нет необходимости дублировать общий код в каждом из подклассов. Общий код может быть реализован в родительском классе и унаследован всеми подклассами.

IS-A («является»)

Давайте рассмотрим пример: есть три класса: *JavaProgrammer*, *PhotonProgrammer*, *JavaScriptProgrammer*. О любом из объектов этих классов можно сказать "является Programmer":

```
каждый JavaProgrammer является Programmer  
каждый PhotonProgrammer является Programmer  
каждый JavaScriptProgrammer является Programmer
```



и т.д., или на английском: *every JavaProgrammer IS A Programmer*

Т.е. каждый объект JavaProgrammer, PhotonProgrammer или JavaScriptProgrammer является подтипом более общего класса Programmer. Такое отношение между классам называют отношением **IS-A**.

Отношение IS-A («является») - основной принцип применения наследования. Любой дочерний класс является подтипом класса родителя или, другими словами, любой дочерний класс **IS-A** родительский класс

Объявление дочернего класса

Что бы объявить дочерний класс используется ключевое слово `extends` :

```
public class JavaProgrammer extends Programmer{  
  
}
```



Переопределение методов (overriding)

Если в дочернем классе объявить метод с тем же именем, возвращаемым типом и параметрами, что и метод в суперклассе, метод дочернего класса переопределит поведение метода родительского класса

Такой метод называется *переопределенным* и может быть помечен как `@Override`

```
public class Programmer{  
    public void writeCode(){  
        System.out.println("write code");  
    }  
}
```



```
public class JavaProgrammer extends Programmer{  
    @Override  
    public void writeCode(){  
        System.out.println("write Java code");  
    }  
}
```



В примере выше `writeCode()` в `JavaProgrammer` - переопределенный метод. Он переопределяет поведение родительского метода для всех `JavaProgrammer`.

Важное

1. Тип переменной определяет доступные методы, тип значения - определяет поведение
2. Любой дочерний класс является подтипом класса родителя (IS-A)