Generics: super, extends, list

Java Collections
5 уровень, 5 лекция

ОТКРЫТА

— Новая интересная тема — wildcards.

По смыслу, это что-то вроде шаблона «*», который совпадает с чем угодно.

Но давай начнем издалека.

Представь, что у тебя есть класс «Warrior(Воин)» и метод, который вычисляет, какой из двух воинов сильнее. Вот как, например, это могло бы выглядеть:

Пример 1

```
1  class WarriorManager
2  {
3   public static boolean fight(Warrior w1, Warrior w2)
4   {
5    return w1.power > w2.power;
6   }
7  }
```

Пример вызова

```
1  MagicWarrior mag = new MagicWarrior();
2  ArcherWarrior archer = new ArcherWarrior();
3
4  boolean isMagicCooler = WarriorManager.fight(mag, archer);
```

MagicWarrior и ArcherWarrior – это классы-наследники от Warrior.

Немного простовато, но для примера сойдет.

— Ок.

— И вот допустим, ты решил сделать аналогичный метод, где уже дерутся воины стенка на стенку.

Пример 1

```
1  class WarriorManager
2  {
3   public static boolean fight(Warrior w1, Warrior w2)
4   {
5    return w1.power > w2.power;
6   }
7
8   public static boolean fight(List<Warrior> w1, List<Warrior> w2)
9   {
10   return …
```

НАЧАТЬ ОБУЧЕНИЕ

Пример вызова

```
ArrayList<MagicWarrior> magi = new ArrayList<MagicWarrior>();
for(int i=0;i<10;i++)
 magi.add(new MagicWarrior());

ArrayList<ArcherWarrior> archers = new ArrayList<ArcherWarrior>();
for(int i=0;i<10;i++)
 archers.add(new ArcherWarrior());

boolean isMagicCooler = WarriorManager.fight(magi, archers); //ошибка компиляции!
```

И вот тут ты встречаешь ошибку компиляции и недоумеваешь, а что не так-то?

Все дело в том, что MagicWarrior – это наследник Warrior и его объекты можно передавать в метод fight(Warrior, Warrior)

А вот List<MagicWarrior> — это не наследник List<Warrior>. И передавать его туда нельзя!

— Как это не наследник?

— А вот так. То List и то – List. Пусть и с параметрами.

— Да, а я как-то сразу не обратил внимание. И что, есть уже решение этой проблемы?

— Ага. Для этого используется более сложная конструкция. Выглядит это так:

Пример 1

```
class WarriorManager
{
 public static boolean fight(Warrior w1, Warrior w2)
 {
  return w1.power > w2.power;
 }

 public static boolean fight(List<? extends Warrior> w1, List<? extends Warrior> w2)
 {
  return …
 }
}
```

«? extends Warrior» обозначает «любой тип, унаследованный от Warrior ».

Т.е. туда можно передать список List<MagicWarrior> и список List<ArcherWarrior>, и все будет работать.

— Так это же отлично. Чем меньше всяких таких проблем, тем лучше.

— Так и я о том же.

— А если мне не нужен наследник Warrior? Что если я хочу, чтобы в метод можно было передать любой List с любым параметром? Так можно?

— Да, для описания этой ситуации существует две записи:

```
List<? extends Object>
```

```
List<?>
```

НАЧАТЬ ОБУЧЕНИЕ

У меня – все.

— Спасибо, Элли, действительно узнал сегодня много нового.

‹ Предыдущая лекция

Следующая лекция ›

+74

Комментарии (34) популярные новые старые

JavaCoder

Введите текст комментария

On1k Уровень 42, Krasnogorsk, Russian Federation 6 июля, 22:51

Интересная отсылка к игре UNO =)

Ответить 0

Наиль Гафиятов Уровень 49 20 апреля, 13:27

Лекция про wildcards, но в итоге даже не написали, что дикая карта это <?>))

А в примере создали метод, принимающий объекты, а передают в него список, ясное дело, что код не будет компилироваться. Амиго неожиданно отупел)

Ответить +4

Darth Nihilus Разработчик спокойствия в **Rage&Flame Industrie** 21 мая 2021, 16:41

Просто жор материи

Ответить +6

Лиза Воренувкина Уровень 43, Кривой Рог, Ukraine 25 сентября 2021, 11:05

это же последние лекции

Ответить 0

Артур Жуков Уровень 32, Астана 16 февраля 2021, 13:13

magi

Ответить +5

LuneFox инженер по сопровождению в **BIFIT** EXPERT 29 января, 20:05

Я до конца лекции надеялся хотя бы раз увидеть нормальное mages. Почему ну luchniki тогда?...

Ответить 0

Дмитрий Рыбин Уровень 40, Краснодар, Россия 2 февраля, 01:05

magi
Википедия
Волхвы были жрецами в зороастризме и более ранних религиях западных иранцев. Самое раннее известное использование слова "волхвы" содержится в трехъязычной надписи, написанной Дарием Великим, известной как надпись Бехистун.

Ответить 0

Valua Sinicyn Уровень 41, Харьков, Украина 15 февраля 2021, 08:56

Для понимания лучше, конечно, использовать определения на английском:

<? extends T> - Upper Bounded Wildcard
<? super T> - Down Bounded Wildcard
<?> - Unbounded Wildcard

НАЧАТЬ ОБУЧЕНИЕ

Евгений Ведущий инженер в **ПАО Сбербанк** EXPERT — 29 июня 2020, 20:45

За 900++ задач так ни разу и не столкнулся с проблемой типизаций списков)
Любопытная штука.

Ответить +4

Сиявуш Android Developer в **Эсхата Банк. Таджики** EXPERT — 28 февраля 2020, 15:35

Generics: super, extends, list - тут только про extends. А про super и list?
<? super ??>

Ответить +7

Vorlock Уровень 31, Днепр, Украина — 26 января 2020, 15:48

про PECS уже было ?

Ответить +1

A.S. Уровень 35 — 26 ноября 2019, 13:34

Пытался вернуть HashMap со списками, долго не понимал почему не проходит валидацию. Для решения надо создать HashMap <K, V> и положить в него соответствующие ключи/значения из аргументов, не сами списки.

Ответить 0

Swizbiz Уровень 37, Москва, Россия — 29 мая 2019, 20:52

Просто оставлю это здесь
YouTube

Ответить +21

Александр Уровень 41, Екатеринбург — 3 сентября 2021, 13:13

Хорошая лекция. Спасибо.

Ответить 0

Показать еще комментарии