

Jazyk Java 1

Seminář 5

Daniel Bazala



Katedra informatiky
Univerzita Palackého v Olomouci

Generické typy

Generické typy

- Generický typ je generická třída nebo rozhraní, které mají datový typ/typy jako parametr/parametry.
- Datový typ určujeme při vytváření instance generické třídy.
- Generické typy umožňují znovu-použít stejný kód pro různé datové typy.
- Generiku hojně využívají např. kolekce

Negenerická verze třídy Box

- nutnost přetypování

```
public class Box {  
    private Object item;  
  
    public void setItem(Object item) {  
        this.item = item;  
    }  
  
    public Object getItem() {  
        return item;  
    }  
}
```

```
Box box = new Box();  
box.setItem(5);  
int i = (int)box.getItem();
```

Generická verze třídy Box

- bez nutnosti přetypování

```
public class Box<T> {  
    // T stands for "Type"  
    private T item;  
  
    public void setItem(T item) {  
        this.item = item;  
    }  
  
    public T getItem() {  
        return item;  
    }  
}
```

```
// Syntaktický cukr pro Box<Integer> box = new Box<Integer>();  
Box<Integer> box = new Box<>();  
box.setItem(5);  
int i = box.getItem();
```

Omezení generických typů

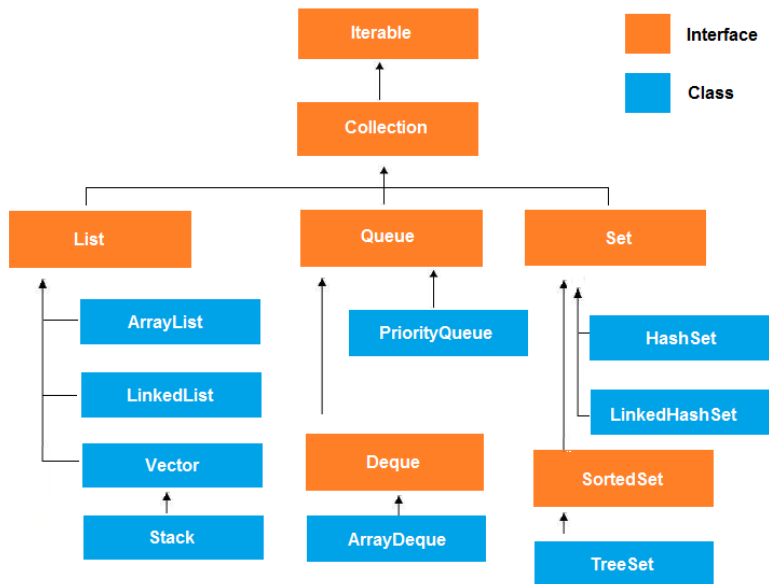
- umožňuje definovat omezení na generické datové typy
- `class Box<T extends Integer> { ... - omezení shora`
- `class Box<T extends Item & Comparable<T>> { ... - dvojí omezení`
- omezení metod:
- `public static void process(List<? extends Foo> list) { ... - omezení shora`
- `public static void addNumbers(List<? super Integer> list) { ... - omezení zdola`

Generické typy

- Více: <https://docs.oracle.com/javase/tutorial/java/generics/types.html>

Kolekce

Kolekce v Javě



Generická kolekce List

- rozhraní `java.util.List`
- <https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/List.html>

```
List<TYP> list;  
List<Integer> list1 = new ArrayList<>();  
list1.add(1);  
list1.add(2);  
System.out.println(list1); // [1, 2]  
List<String> list2 = new LinkedList<>();  
list2.add("Pavel");  
list2.add("Jirka");  
System.out.println(list2); // [Pavel, Jirka]  
System.out.println(list2.get(1)); // Jirka  
list2.remove(0);  
System.out.println(list2); // [Jirka]  
list2.remove("Jirka");  
System.out.println(list2); // []
```

Rozhraní Iterable

- <https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/Iterable.html>
- metoda `Iterator<T> iterator()`

Rozhraní Collection

- <https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/Collection.html>
- metody `boolean add(E e)`, `boolean remove(Object o)`, `void clear()`, `boolean equals(Object o)`, `boolean contains(Object o)`, `int size()`, `Object[] toArray()`, `<T> T[] toArray(T[] a)`, ...

Rozhraní Comparable

- <https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/Comparable.html>
- rozhraní implementují objekty, které mají být porovnatelné
- využívá se u třízení

Mapy

- rozhraní pro ukládání položek typu <klíč, hodnota>
- mapa nemůže obsahovat duplicitní klíče
- pro každý klíč se mapuje pouze na jednu hodnotu

```
Map<Integer, String> hashMap = new HashMap<Integer, String>();
```

```
hashMap.put(7, "Peter");  
hashMap.put(5, "Philip");  
hashMap.put(2, "Martin");
```

```
System.out.println(hashMap.get(2)); // Martin
```

- Více: <https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/Map.html>

Úkol

Úkol seminář 5

- <http://marcus.webly3d.net/ukol5>