Java Advanced

Week 2: Generics

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Wat is generics?

```
public class GenericBox<T> {
    private T item;

    public T getItem() {
        return item;
    }

    public void setItem(T item) {
        this.item = item;
    }
}
```

Using Generics is all about abstracting the type from a method or a class to create generic methods or classes applicable to more type than a specific type.

Wat is generics?

```
GenericBox<String> stringBox = new GenericBox<>();
stringBox.setItem("elephant");
System.out.println(stringBox.getItem());

GenericBox<Long> longBox = new GenericBox<>();
longBox.setItem(123569L);
System.out.println(longBox.getItem());

GenericBox<Person> personBox = new GenericBox<>();
personBox.setItem(new Person("Mickey Mouse", 55));
System.out.println(personBox.getItem().getName());
```

Generieke klasse: type inperken

```
public class Habitat<A extends Animal> {
    private A inhabitant;

    public void addInhabitant(A inhabitant) throws
                                        HabitatOvercrowdedException {
        if (inhabitant != null) {
            throw new HabitatOvercrowdedException("Habitat full.");
        }
        this.inhabitant = inhabitant;
    }
}
```

Generieke klasse: type inperken

```
public class Demo2 {
    public static void main(String[] args) throws
                                                HabitatOvercrowdedException {
        //Habitat<String> stringHouse = new Habitat<>();
        Habitat<Snail> cochlea = new Habitat<>();
        //cochlea.addInhabitant(new Cat());
        cochlea.addInhabitant(new Snail());
    }
}
```

Generieke interfaces en ArrayList

```
public interface Habitat<A extends Animal> {
        void addInhabitant(A inhabitant);

        void removeInhabitant(A inhabitant);

        A catchInhabintant();

        int getNumberOfInhabitants();
}
```

Generieke interfaces en ArrayList

```
public class Aquarium implements Habitat<Fish> {
    private static final Random RANDOM = new Random();
    private List<Fish> fishes = new ArrayList<>();

    @Override
    public void addInhabitant(Fish inhabitant) {
        fishes.add(inhabitant);
    }

    @Override
    public void removeInhabitant(Fish inhabitant) {
        fishes.remove(inhabitant);
    }

    @Override
    public Fish catchInhabintant() {
        return fishes.get(RANDOM.nextInt(getNumberOfInhabitants()));
    }

    @Override
    public int getNumberOfInhabitants() {
        return fishes.size();
    }
}
```

Generieke methoden

```
public class AnimalUtility {
    public static <E extends Animal> void moveAnimals(
     E animalToMove, Habitat<E> fromHabitat, Habitat<E> toHabitat) {
        fromHabitat.removeInhabitant(animalToMove);
        toHabitat.addInhabitant(animalToMove);
    }

    public static boolean possibleProblem(Habitat<?> habitat) {
        for (int i = 0; i < 5; i++) {
            Animal animal = habitat.catchInhabintant();
            if (animal.isSick()) {
                return true;
            }
        }
        return false;
    }
}
```

Generieke methoden

```
public class GenericBoxUtility {
    public static <T extends Animal> boolean isSick(
                                              GenericBox<T> animalBox) {
        return animalBox.getItem().isSick();
    }

    public static String displayBox(
                            GenericBox<? super Animal> surpriseBox) {
        return surpriseBox.getItem().getClass().getSimpleName();
    }
}
```

Achter de schermen

- Generics don’t exist at runtime!
- Er bestaat slechts 1 klasse-bestand van een generieke klasse.
- Alle informatie mbt het datatype is weggehaald in de gecompileerde klasse: type erasure
- Generieke type van de klasse is dus niet toegelaten in static variabelen
- Instanceof kan niet gebruikt worden om te controleren of een object van een gegeven generieke klasse is

Achter de schermen

```
public class GenericBox<T> {
    private static T something;
    private T item;
    private T[] items;

    public GenericBox(){
        item = new T();
        items = new T[10];
    }
```