Föreläsning 9
Objektorienterad programmering
DD1332

- Serialization
- Egna generiska datatyper
- Nätverk Socket

1

Serialization

- Lagra objekt på disk eller skicka objekt över nätet
- Endast instansvariabler
- T.ex: `class HuffmanTree implements Serilaizable{…}`
- Läs objektet från fil m.h.a `ObjectInputStream`
- Skriv objektet på fil m.h.a `ObjectOuputStream`

2

Lagra objekt på fil

```
class Node implements Serializable{…}
class SearchTree implements Serializable{
       private Node root;
…
}
class Main{
  public satic void main(String[] args)throws IOException{
    st=new SearchTree();
    //här kommer att finnas kod som bygger trädet
    FileOutputStream fileOut=new FileOutputStream("tree.ser");
    ObjectOutputStream out = new ObjectOutputStream(fileOut);
    out.writeObject(st);
    out.close();
    fileOut.close();
}
```

3

Hämta objekt från fil

```
class Main2{
  public static void main(String[] args)throws IOException{
    SearchTree st = null;
    FileInputStream fileIn = new FileInputStream("tree.ser");
    ObjectInputStream in = new ObjectInputStream(fileIn);
    try{
       st = (SearchTree) in.readObject();
    }catch(ClassNotFoundException cnfe){
       System.err.println("Hittar inte filen SearchTree.class");
       }
    in.close();
    fileIn.close();
    …
}
```

4

Egna generiska typer

- Så här deklarerar man en klassparameter, T, för att sedan använda den i klassdefinitionen:

```
class Klassnamnet<T>{
    T t;
    public klassnamn(T t){
          this.t=t
    }
}
```

- Nu kan **T** vara vilken datatyp som helst vid instansering av klassen `Klassnamnet`

5

Exempel: Uthyrning

- Anta att vi vill deklarera en generell klass som kan användas till uthyrning av olika saker, t.ex Biluthyrning, bostadsuthyrning osv…

6

Exempel: Arv variant

```
public class Uthyrning {
     private List uthyrningsLista;
     private int max;
     public Uthyrning(int maxNum, List ul) {…}
     public Object lamnaUt() {…}
     public void taEmot(Object o) {…}
}
```

7

Exempel: Arv variant

```
class Biluthyrning extends Uthyrning {
  public Biluthyrning(int maxNum, List<Bil>
bilar) {…}
  public Bil lamnaUt() {…}
  public void taEmot(Bil b) {super.taemot(b);}
  public void taEmot(Object o) {
     if (o instanceof Bil) {
       super.taEmot(o);
     } else
        System.out.println("det gar inte att
lagga till!");

   }
```

8

Exempel: Arv variant

```
public static void main(String[] args){
       List<Bil> BilLista = new ArrayList<Bil>();
       BilLista.add(new Bil("Mazda"));
       BilLista.add(new Bil("Volvo"));
       Biluthyrning bilUthyrning = new
             Biluthyrning(2, BilLista);
       Bil bil = bilUthyrning.lamnaUt();
       System.out.println(bil.marke);
       bilUthyrning.taEmot(bil);
       bil = bilUthyrning.lamnaUt();
       System.out.println(bil.marke);
       bilUthyrning.taEmot(bil);
}
```

9

Generisk variant

```
public class UthyrningGenerisk<T> {
  private List<T> uthyrningsLista;
  private int max;
  public UthyrningGenerisk(int maxNum, List<T> ul) {
        this.max = maxNum;
        this.uthyrningsLista = ul;
     }
     public T lamna() {
        return uthyrningsLista.remove(0);
     }
     public void taEmot(T item) {
        uthyrningsLista.add(item);
     }
}
```

10

Generisk variant

```
class TestaUthyrning {
    public static void main (String[] args) {
       List<Bil> BilLista = new ArrayList<Bil>();
       BilLista.add(new Bil("Mazda"));
       BilLista.add(new Bil("Volvo"));
       UthyrningGenerisk<Bil> bilUthyrning = new
                   UthyrningGenerisk<Bil>(2, BilLista);
       Bil bil = bilUthyrning.lamna();
       System.out.println(bil.marke);
       bilUthyrning.taEmot(bil);
       bil = bilUthyrning.lamna();
       System.out.println(bil.marke);
       bilUthyrning.taEmot(bil);
    }
}
```

11

Egna generiska metoder

Om man inte vill att klassen ska ha någon klassparameter däremot metoderna ska vara generiska då deklarerar man <T> i metodhuvudet:

```
class KlassNamnet{
  public <T> void enMetod(T t){…}
  public <T extends Fordon> void annanMetod(T t){…}
}
```

12