

EDAA45 Programmering, grundkurs

Läsvecka 13: På begäran

Björn Regnell

Datavetenskap, LTH

Lp1-2, HT 2016

13 På begäran

- Grumligt-lådan
- Nyfiken-på-lådan
- Repetition diverse

Grumligt-lådan

Översikt av innehållet i Grumligt-lådan

Ämnen (antal)

- arv (4)
- getter, setter (4)
- kompanjonsobjekt (4)
- case-objekt (4)
- try (4)
- java (3)
- matriser (3)
- sortering (3)
- ArrayBuffer (2)
- loopar (2)
- Map och map (2)
- option (2)
- problemlösning (2)
- (1)
 - funktionsvärden; generiska funktioner; groupBy; in-mixning; klasser och case-klasser; konstanter; konstruktor; läsa från textfil; läsa kod; match case; objektfabriksmetod; pirateslabben; private[this]; sortBy; static; type; typparameter;

Nyfiken-på-lådan

Översikt av innehållet i Nyfiken-på-lådan

Ämnen (antal)

- gränssnitt (3)
- Java (3)
- rekursion (3)
- funktionsprogrammering (2)
- generiska typer (2)
- implicit (2)
- trådar, Future (2)
- webb, html (2)
- (1)
 - bilder, ljud och spara filer; enkel AI; minneshantering i olika språk (GC eller manuell); gå igenom och förklara QuickRef mer noggrant; hacka andras kod i låsta applikationer; hashcode; kryptering; prestanda och minnesåtgång Scala vs Java; Stream[T]; teorin bakom neurala nätverk;

Repetition diverse

Oföränderlig punkt i Scala och Java

I Scala: (utan case-klass ingen najs toString och måste skriva **new**, etc.)

```
class Point(val x: Int, val y: Int)
```

I Java:

```
public class JPoint {  
    private int x;  
    private int y;  
  
    public JPoint(int x, int y){  
        this.x = x;  
        this.y = y;  
    }  
  
    public int getX(){  
        return x;  
    }  
  
    public int getY(){  
        return y;  
    }  
}
```


Föränderlig punkt i Scala och Java

I Scala:

```
class Point(var x: Int, var y: Int)
```

I Java:

```
public class JPoint {  
    private int x;  
    private int y;  
  
    public JPoint(int x, int y){  
        this.x = x;  
        this.y = y;  
    }  
  
    public int getX(){  
        return x;  
    }  
  
    public int getY(){  
        return y;  
    }  
  
    public void setX(int x){  
        this.x = x;  
    }  
  
    public void setY(int y){  
        this.y = y;  
    }  
}
```

Punkt med räknare och setter i Scala

Övning: lägg till getter och setter för y-koordinaten.

```
class Point(private var myX: Int, private var myY: Int){  
  import Point._  
  def x = myX  
  def x_=(newX: Int): Unit = {  
    myX = newX  
  }  
  myCount += 1 // kod i klasskroppen körs vid konstruktion  
}  
  
object Point {  
  private var myCount = 0  
  def count = myCount  
}
```

Man brukar kalla privata attribut som har getter (och ev. setter) för något i stil med myX eller vanligare _x för att namnet inte ska krocka med getter/setter.

Punkt med räknare i Java

```
public class JPoint {  
    private int x;  
    private int y;  
  
    static private int count = 0; // static: finns bara en upplaga av detta attribut  
  
    public JPoint(int x, int y){  
        this.x = x;  
        this.y = y;  
        count++;  
    }  
  
    public int getX(){  
        return x;  
    }  
  
    public int getY(){  
        return y;  
    }  
  
    public void setX(int x){  
        this.x = x;  
    }  
  
    public void setY(int y){  
        this.y = y;  
    }  
  
    static public int getCount(){  
        return count;  
    }  
}
```

Robot

Följ Robot-exemplet från förra årets Java-kurs här:

<https://github.com/bjornregnell/lth-eda016-2015/blob/master/lectures/notes/week05.pdf>

Övning: översätt till Scala.