6)

```
public class Dog {
	private String dogName;
	public Dog(String dogName){
		/*
		 * Sätt Dog's name-variabel till samma som name-parametern
		 */
		this.dogName = dogName;
	}
}
```

7)

```
public class RussianDoll {
	private int nbrOfContainedDolls;
	public RussianDoll(){
		nbrOfContainedDolls=7;
	}
	public boolean isCore(){
		return nbrOfContainedDolls==0;
	}
	public void removeOuterLayer(){
		nbrOfContainedDolls--;
	}
	public String getNbrOfLayers(){
		return "This number is secret! Use rekursion-method " +
				"instead to find out:)";
	}
	public static int rekursiveDollCounter(RussianDoll doll, int
countedDolls){
		if(!doll.isCore()){
			/*
			 * Om vi inte är vid kärnan så öka räknaren och ta
bort ett lager
			 */
			doll.removeOuterLayer();
			return rekursiveDollCounter(doll, countedDolls + 1);
		}
		else{
			//Lägg till "kärn-dockan" och returnera
			return countedDolls;
		}
	}
	public static void main(String[] args){
```

```
            RussianDoll russianDoll1 = new RussianDoll();
            System.out.println(rekursiveDollCounter(russianDoll1, 0));
      }
}
```

8)

```
public static int iterativeDollCounter(RussianDoll doll){
            int countedDolls = 0;
            while(!doll.isCore()){
                  countedDolls++;
                  doll.removeOuterLayer();
            }
            return countedDolls;
      }

//Test av metoderna
public static void main(String[] args){
      RussianDoll russianDoll1 = new RussianDoll();
      System.out.println(rekursiveDollCounter(russianDoll1, 0));
      RussianDoll russianDoll2 = new RussianDoll();
      System.out.println(iterativeDollCounter(russianDoll2));
}
```

Kämpa på med kodandet, annars kommer Crazy Hipster Cat och tar er;)
//Anton

tinyurl.com/crazyhipstercat