

Stäng av datorerna och skriv allt med papper och penna idag.

1)

Ta ett nytt papper. Samarbeta i gruppen runt ert bord och skriv en klass som representerar en Chalmers-student.

- A. Tänk ut vilka egenskaper (klassvariabler och instansvariabler) som studenten ska ha och skriv kod för dessa.
- B. Tänk ut vilka beteenden (metoder) som studenten ska ha och skriv kod för dessa.
- C. Tänk ut hur studenten ska skapas och skriv en eller flera konstruktorer för detta.
- D. Ge variabler och metoder förklarande namn så att det blir lätt att förstå (kod-kommentarer ska inte behövas).
- E. När ni är nöjda med klassen ger ni bort pappret till bordet till höger om er samtidigt som ni får pappret från bordet till vänster.

2)

- A. Gå igenom klassen ni just har fått tillsammans och se om ni förstår.
- B. Skriv en testklass med en main-metod som testar att metoderna i student-klassen gör vad de ska och att variablerna är vad de borde vara.
- C. Skicka därefter klassen vidare till bordet till höger samtidigt som ni tar emot klassen från bordet till vänster.

3)

- A. Skriv javadoc för klassen ni just fått samt för alla dess metoder.
 - o Använd t.ex. taggarna:
 - @param
 - @return
 - @throws
- B. Skicka därefter vidare pappret en sista gång.

4)

Titta på klassen ni just fått. Skriv nu en WorkingStudent-klass som extendar/ärver Student-klassen ni fått. Det nya WorkingStudent-objektet som ska kunna skapas ska till exempel ha variabler och metoder för:

- Nuvarande företag som hen jobbar på
- Lön
- Arbetslivserfarenhet i antal år

SI-Pass 5

tinyurl.com/sipass5

5)

Titta på klassen nedan:

```
public class Cat {
    private static int nbrOfLegs = 4;
    private String name;
    public Cat(String catName){
        name = catName;
    }
    public static int getNbrOfLegs(){
        return nbrOfLegs;
    }
    public String getName(){
        return name;
    }
}
```

a)

Titta på denna testklass för katten:

```
public class TestCat {
    public static void main(String[] args){
        //Create cats
        Cat cat1 = new Cat("Sven");
        Cat cat2 = new Cat("Nils");

        //Test names
        String cat1Name = cat1.getName();
        String cat2Name = cat2.getName();
        System.out.println("Name of cat1: " + cat1Name);
        System.out.println("Name of cat2: " + cat2Name);

        //Test number of legs
        int numberOfLegs = Cat.getNbrOfLegs();
        System.out.println("The number of legs is: " +
        numberOfLegs);
    }
}
```

b)

Varför skriver man en katt-instans framför anrop av getName()?

SI-Pass 5

tinyurl.com/sipass5

c)

Varför skriver man Cat framför anrop av getNbrOfLegs()?

d)

Förklara vad varje ord nedan betyder:

- **private String name;**
 - private:
 - String:
 - int:
- **private static int nbrOfLegs = 4;**
 - private:
 - static:
 - int:
 - nbrOfLegs:

6)

Ofta vill man för tydlighetens skull ha samma namn på en variabel som på parametern som skickas in. Hur gör man då? Fyll i koden som saknas nedan:

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

SI-Pass 5

tinyurl.com/sipass5

7)

Tänk er att ni har en rysk docka där ni vill ta reda på hur många dockor den består av totalt.



<http://www.ctrlaltnkill.org/2010/12/24/double-fines-next-game-wants-you-to-play-with-dolls>

Klassen för en docka ser ut såhär:

```
public class RussianDoll {
    private int nbrOfContainedDolls;
    public RussianDoll(){
        nbrOfContainedDolls=<SECRET>;
    }
    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:>";
    }
}
```

Som ni ser är antalet dockor i dockan **hemligt**! För att ta reda på detta skriver ni istället en **rekursiv metod** som slutligen ska returnera totala värdet på antalet dockor. Använd er av metoderna: **public boolean isCore()** och **public void removeOuterLayer()**

8)

Kan man även lösa uppgift 7 iterativt? Hur i så fall?

OOP

Handledare: Anton Palmqvist
palmqvist.anton(SnabelA)gmail.com

SI-Pass 5

tinyurl.com/sipass5

Glöm inte att gå med i **SI-gruppen** på **facebook**!



Sök på "**SI - Programmers paradise**" på facebook eller använd länken:
<https://www.facebook.com/groups/726850060722201/>