

# Inlämning 3

## Objektorienterad programmering med Java

Emil Björklund - embj3739  
emilbjorklund@live.com

March 20, 2019

# Inledning

- IDE: Visual Studio Code
- Java version: JDK 8 update 191
- OS: MacOS Mojave

**Note.** Källkoden bifogas i separat .zip som finns i inlämningen.

## Struktur

### Class Window

Denna klass innehåller main konstruktör och anropas därmed av användaren vid start. Klassen öppnar en "JFileChooser" för att användaren enkelt skall kunna importera önskade bilder. Om användaren inte importerar ett jämnt antal bilder får användaren på nytt importera korrekt antal. Ett jämnt antal bilder krävas för att programmet skall kunna applicera en framsida samt baksida av varje kort. När användaren har lyckats med sin importering av bilder sparas alla i en array av "Files". Denna array skickas vidare när main deklarerar ett nytt "Window" objekt.

I objektet "Window" skapas en "Jframe" samt ett objekt kallat Panel där all hantering av bilder och position sker. Objektet panel läggs sedan till i "Jframe".

### Class Panel

I klassen Panel deklareras det två arrayer. Den första arrayen är en tom "Image" array med lika många platser som "Files" arrayen från main är. Den andra arrayen är av typen boolean som innehåller information om respektive bild visar baksida eller ej. När arrayerna är deklarerade anropas metoden "initLayers" som ser ut som följande:

```
private void initLayers(File files[]) {  
    for (int i = 0; i <= (files.length - 1)/2; i++) {  
        layers[i] = new Image(i*10,i*20,files[i],files[i+(files.length)/2]);  
    }  
}
```

Denna loop itererar längden av "Files" arrayen. Varje position i arrayen fylls med ett Objekt av klassen Image. Denna klass tar även hand om events som "MouseClicked", "MouseDraged", "MousePressed" för att kunna interagera med bilderna i programmet. Logiken bakom "MouseClicked" och "MouseDraged" är nästan lika och ser ut som följande:

```
public void mousePressed(MouseEvent e) {  
    for (int i = 0; i <= layers.length - 1; i++) {  
        if (layers[i].constraint(e.getX(), e.getY())){  
            Image temp = layers[i];  
  
            for (int j = i; j > 0; j--){  
                layers[j] = layers[j-1];  
                isBackside[j] = isBackside[j-1];  
            }  
        }  
    }  
}
```

```

        layers[0] = temp;
        isActive = true;
        break;
    }
}
}

```

När musen trycks ner eller klickas loopar en for-loop från noll till längden av bild arrayen. Om något lagers bild omsluter kordinaterna för musen flyttas indexeringen om i arrayen så denna bild hamnar längst fram. För att kolla om musen är över en bild anropas metoden "constraint" i klassen "Image" När lagerna har bytt plats så att den aktiva bilden är längst fram i arrayen blir variabeln "isActive" sann och möjliggör för eventet "MouseDraged" att anropa klassen "Images" metod action. Efter anropet kallas funktionen "repaint". I klassen "Panel" finns metoden "paintCompoent" denna metod itererar arrayen för bilder baklänges så att den aktiva bilden målas upp sist.

## Class Image

Klassen Image har all information om bilden och dess position. När klassen deklarerats som ett objekt från klassen Panel anropas konstruktor metoden "readImage" denna metod läser in två bilder från sökvägen som ges som array i metod-huvudet. Koden kan ses nedan.

```

public void readImage(File file1,File file2) {
    try
    {
        image = new BufferedImage(100, 100,BufferedImage.TYPE_INT_ARGB);
        image_backside = new BufferedImage(100, 100,BufferedImage.TYPE_INT_ARGB);
        image = ImageIO.read(file1);
        image_backside = ImageIO.read(file2);

        System.out.println("Reading complete.");
    }
    catch(IOException e)
    {
        System.out.println("Error: "+e);
    }
}

```

Klassen har även variabler som har information om bildens position uttryckt i x och y koordinater. Om ett mus event händer i klassen Panel anropas klassen Images metod constraint för att verifiera om musens koordinater är inom bildens. Vidare har klassen även en metod där bildens koordinater uppdateras beroende på musens koordinater för att sedan kunna ritas om.