

## Interaktion mellan objekt

Att skapa interagerande objekt



## En digital klocka

11:03



## Abstraktion och modularisering

- Abstraktion är förmågan att bortse från detaljerna och i stället fokusera på ett problems viktigaste delar.
- Modularisering handlar om att dela upp en helhet i väldefinierade delar som kan beskrivas var för sig och som kan interagera på ett väldefinierat sätt.



## Att modularisera klockans visare

11:03

En fyrsiffrig visare?

Eller två  
tvåsiffriga visare?

11

03



## Implementation - NumberDisplay

```
public class NumberDisplay
{
    private int limit;
    private int value;

    Constructor and
    methods omitted.
}
```



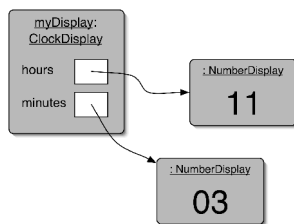
## Implementation - ClockDisplay

```
public class ClockDisplay
{
    private NumberDisplay hours;
    private NumberDisplay minutes;

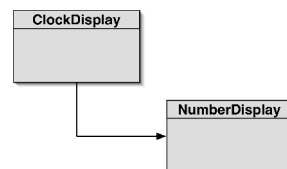
    Constructor and
    methods omitted.
}
```



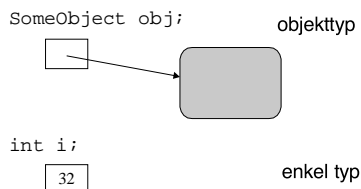
## Objektdiagram



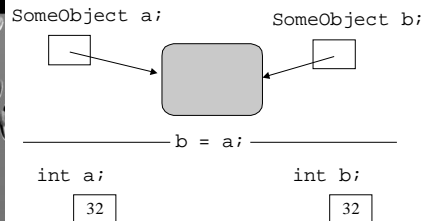
## Klassdiagram



## Enkla typer vs. objekttyper



## Enkla typer vs. objekttyper



## Källkoden: NumberDisplay

```
public NumberDisplay(int rolloverLimit)
{
    limit = rolloverLimit;
    value = 0;
}

public void increment()
{
    value = (value + 1) % limit;
}
```



## Källkoden: NumberDisplay

```
public String getDisplayValue()
{
    if(value < 10)
        return "0" + value;
    else
        return "" + value;
}
```



## Objekt som skapar objekt

```
public class ClockDisplay
{
    private NumberDisplay hours;
    private NumberDisplay minutes;
    private String displayString;

    public ClockDisplay()
    {
        hours = new NumberDisplay(24);
        minutes = new NumberDisplay(60);
        updateDisplay();
    }
}
```



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling  
Svenska versionen av Eric Astor och Jacek Malec

13

## Metodanrop

```
public void timeTick()
{
    minutes.increment();
    if(minutes.getValue() == 0) {
        // it just rolled over!
        hours.increment();
    }
    updateDisplay();
}
```



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling  
Svenska versionen av Eric Astor och Jacek Malec

14

## En intern metod

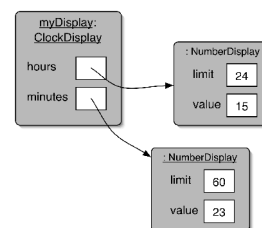
```
/**
 * Update the internal string that
 * represents the display.
 */
private void updateDisplay()
{
    displayString =
        hours.getDisplayValue() + ":" +
        minutes.getDisplayValue();
}
```



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling  
Svenska versionen av Eric Astor och Jacek Malec

15

## Objektdiagrammet för ClockDisplay



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling  
Svenska versionen av Eric Astor och Jacek Malec

16

## Objekt som skapar objekt

I klassen NumberDisplay:

```
public NumberDisplay(int rollOverLimit);
```

formell parameter

I klassen ClockDisplay:

```
hours = new NumberDisplay(24);
```

aktuell parameter



## Metodanrop

- Interna metodanrop

```
updateDisplay();  
...  
private void updateDisplay()
```

- Externa metodanrop

```
minutes.increment();  
objekt.metodNamn(parameterlistan)
```



## this

```
public class MailItem  
{  
    private String from;  
    private String to;  
    private String message;  
  
    public MailItem(String from, String to, String message)  
    {  
        this.from = from;  
        this.to = to;  
        this.message = message;  
    }  
    ...  
}
```




## Inlämningsuppgift 1

Att tänka på:

- Korrekt kod syntaktiskt och semantiskt.
- Skall kompilera utan fel.
- Korrekta kommentarer.
- Bra val av identifierarnamn.
- Indentering av koden!

Ledning: Använd de konventioner som boken förmedlar!





## Viktiga begrepp

- abstraction
- modularization
- classes define types
- class diagram
- object diagram
- object references
- primitive types
- object types
- object creation
- overloading
- internal/external method call
- debugger

