Skolan för Datavetenskap och kommunikation

Programmeringsteknik

Föreläsning 9

ARV

- Läs kap 9 i boken!

arv = *inheritance*

kodåtervinning

- Samma satser om igen?
  - *Skriv en slinga eller en funktion!*
- Samma funktioner i nytt program?
  - *Skriv en modul och importera den!*
- Samma data som skickas in i alla funktionerna?
  - *Skriv en klass!*
- Flera klasser med liknande attribut och metoder?
  - *Skriv en superklass och låt klasserna ärva från den!*

Arv

- *Arv* låter oss återanvända klasser.
- Exempel: `class Fetknapp(Button):`
- Klassen Fetknapp ärver alla
  - attribut
  - metoder

  från klassen Button.
- Fetknapp är *subklass* till Button
- Button är *superklass* till Fetknapp

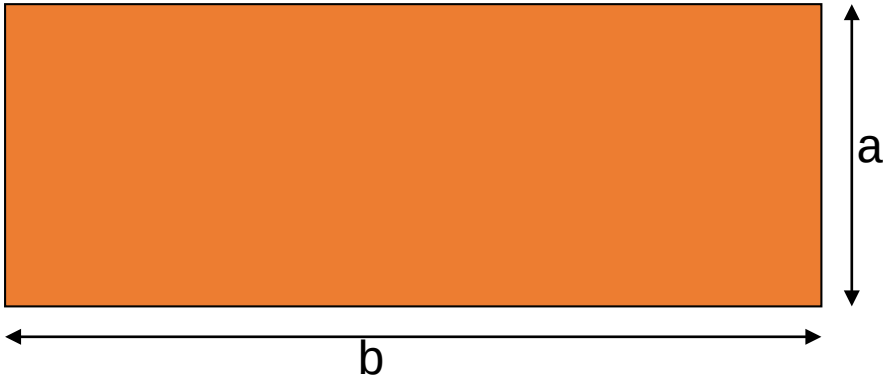
Fetknapp

```
class Fetknapp(Button):

    def snyggaTill(self):
        knapp.config(height = 3, width = 9, \
                     font = ('times', 32, 'bold'))


roten = Tk()
knapp = Fetknapp(roten, text="Tryck inte", \
command=byttext)
knapp.snyggaTill()
```

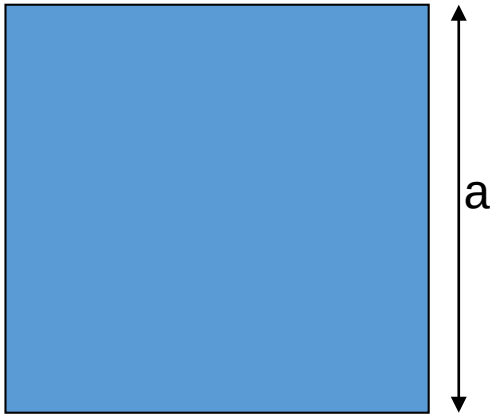
Exempel: Geometri

- Ett enkelt exempel som visar hur arv fungerar:
  - Parallellogram är den mest generella figuren -den får bli superklass
  - Rektangel är en sorts parallellogram med räta vinklar - vi låter den vara subklass till Parallellogram
  - Kvadrat är en sorts rektangel med lika sidor - den får vara subklass till Rektangel

Parallellogram:
area = h*b
omkrets = 2*(a+b)



Rektangel:
area = a*b
omkrets = 2*(a+b)



Kvadrat:
area = a*a
omkrets = 4*a

```
# Modul med geometri-klasser
#*********************************************************
#************** superklassen Parallellogram **************
#*********************************************************

class Parallellogram(object):

    def __init__(self,a,b,h):
        self.kant1 = a
        self.basKant = b
        self.hojd = h

    def area(self):
        return self.hojd*self.basKant

    def omkrets(self):
        return 2*(self.kant1+self.basKant)
```

```
#*************************************************************
#******  Rektangel är subklass till Parallellogram ****
#*************************************************************
class Rektangel(Parallellogram):

    def __init__(self,a,b):
        self.kant1  =  a
        self.basKant = b

    def area(self):
        return self.kant1*self.basKant
```

```
#*************************************************************
#*********  Kvadrat är subklass till Rektangel ***********
#*************************************************************
class Kvadrat(Rektangel):

    def __init__(self,a):
        self.kant1 = a
        self.basKant = a
```

Kvadrat-objekt

- Vi skapar en instans av Kvadrat…

  … och anropar area() och omkrets().
- Vilka metoder är det som används?

```
from geometri import *

def main():
    sida = input("Hur stor är sidan på din kvadrat? ")
    kvadrat = Kvadrat(sida)
    print("Din kvadrat har arean", kvadrat.area())
    print("och omkretsen", kvadrat.omkrets())

main()
```

moduler

- En modul kan man hämta till programmet med en import-sats
- Jämför

  ```
  import random
  from random import *
  ```

- En egen fil med klasser och funktioner kan importeras på samma sätt

  ```
  from geometri import *
  ```