

Föreläsning 7 Programmeringsteknik DD1310

- Egen modul
- Metoderna: `__str__`, `__lt__`, ...
- Meddelande
- Arv, Överriding av metoder, Polymorfism

Översikt
Egen modul
__str__, __lt__, ...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Egen modul

- Man kan samla alla funktioner och klasser i en fil och importera filen i huvudprogrammet, precis på samma sätt som man importerar moduler från standard biblioteket.

Översikt
Egen modul
__str__, __lt__, ...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

metoderna

`__str__`, `__lt__`...

- `__str__(self)`
Används om man vill få ut en textrepresentant av ett objekt
- `__lt__`, `__gt__`, `__le__`, `__ge__`, `__eq__` och `__ne__`
Används om man vill jämföra två objekt (oftast av samma typ) med följande jämförelseoperatorer: `<`, `>`, `<=`, `>=`, `==` och `!=`

Oversikt
Egen modul
__str__, __lt__,...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Publika och privata metoder

- Om en metod definieras med ett namn som börjar med två understreck då är metoden privat. Alltså metoden kan endast anropas inom klassen och inte utanför klassen.
- Om ett metodnamn inte definieras med två understreck då är metoden publik. Alltså metoden kan anropas överallt.

Oversikt
Egen modul
__str__, __lt__,...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Meddelande

- När objekt A anropar en metod hos objekt B då skickar A ett meddelande till B.

Oversikt
Egen modul
__str__, __lt__,...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Exempel

```

class Student(object):
    def __init__(self,n):
        self.__name = n
    def getName(self):
        return self.__name

class Teacher(object):
    def __init__(self,n):
        self.name = n
    def introduce(self,s):
        theBestStudent = s.getName()
        print("Mitt namn är",self.name,"och är stolt över min elev", theBestStudent)

s=Student("Eva Matsson")
t=Teacher("Malin Samuelsson")
#I följande kod skickas ett meddelande från teacher objektet, t, till student objektet, s.
t.introduce(s)

```

Översikt
Egen modul
__str__, __lt__, ...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Arv

- I objektorienterade språk finns arvmöjlighet.
- Subklass** ärver från **superklass**.
- En subklass ärver alla metoder och attribut från superklassen.

Översikt
Egen modul
__str__, __lt__, ...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Exempel

```

import math
class Ellips:
    def __init__(self, a, b):
        self.lang = a
        self.kort = b

    def area(self):
        return math.pi * self.lang * self.kort

class Cirkel(Ellips):
    def __init__(self, r):
        super(Cirkel, self).__init__(r, r)

    def omkrets(self):
        return 2 * math.pi * self.lang

```

Översikt
Egen modul
__str__, __lt__, ...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Överriding av metoder

- Överriding av en metod innebär definiera en metod i subklassen som redan är definierad i superklassen.

Oversikt
Egen modul
__str__, __lt__,...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Exempel

```

class Parallelogram:
    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)

```

Oversikt
Egen modul
__str__, __lt__,...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Exempel

```

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

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

class Kvadrat(Rektangel):
    def __init__(self, a):
        super(Kvadrat, self).__init__(a, a)

```

Oversikt
Egen modul
__str__, __lt__,...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Utökning

- Man utökar en subklass om man i klassen lägger till metoder eller attribut som inte är deklarerade i superklassen. Detta är användbar när man vill skapa en mer specialiserad version av superklassen.

Oversikt
Egen modul
__str__, __lt__,...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Polymorfism

- Polymorfism är möjligheten att kunna skicka samma meddelande till objekt av olika klasser och få olika resultat.

Oversikt
Egen modul
__str__, __lt__,...
Publik/Privat
Meddelande
Arv
Överriding
Utökning
Polymorfism
Sammanfattning

Sammanfattning

- Dela upp ditt program till olika filer och använd moduler för att införa funktioner och klasser
- `__str__` är en metod som anropas när man använder objektet i samband med `print`
- `__lt__` är en metod som anropas när man använder objekt i jämförelseoperatorer, användbar i samband med funktionen `sort()`
