

## PROGRAMMERINGSTEKNIK

FÖRELÄSNING 11

## FÖRELÄSNING 11

- Exception
- Felhantering med try...except
- Klasserna Spelkort, Hand, Lek - exempel med
  - arv
  - objekt i lista
  - nästade slingor
  - random.shuffle()
  - defaultparameter

## EXCEPTION

- Fel att öppna en fil för läsning om den inte finns.
- Felet heter IOError:
 

```
>>> fil = open("kudde.txt","r")
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in ?
    fil = open("kudde.txt","r")
IOError:[Errno 2] No such file:
'kudde.txt'
```

## OLIKA FELTYPER

Exception	När uppkommer det?
<code>IOError</code>	Om man försöker öppna en fil som inte finns.
<code>SyntaxError</code>	Om man skrivit programkod som Python inte kan tolka.
<code>KeyError</code>	Om man försöker använda en nyckel som inte finns i en ordlista.
<code>NameError</code>	Om man använder en variabel utan att ha gett den ett värde .

## TRY-EXCEPT

- Man kan ta hand om fel genom att införa try-except-satser. Exempel:
- ```
def öppna (namn) :
    """Öppna en fil för läsning."""
    try:
        fil = open (namn)
        return fil
    except (IOError) :
        print ("Kan inte öppna filen",namn)
```

## MED SLINGA

- Man kan ta hand om fel genom att införa try-except-satser. Exempel:
- ```
filSaknas = True
while filSaknas:
    try:
        fil = open (namn)
        return fil
    except (IOError) :
        print ("Kan inte öppna filen",namn)
        namn = input ("Ge nytt filnamn: ")
```

## UPPGIFT: VARFÖR?

- Utan try...except går det inte att läsa från fil.
- Programmet går snabbare
- Man slipper stänga filen med close
- Man kan få felutskrift på svenska

## SPELKORT

- Klassen Kort representerar ett spelkort



- Klassens attribut är `valor` (2-10, knekt, dam, kung, ess), `farg` (Klöver, Ruter, Hjärter och Spader) och `framsidanUpp` (True om kortet är uppvänt)
- Metoderna är `__init__` (konstruktorn), `__str__` (kortet som sträng), `vand` (vänder)

```
class Kort(object):
    """ Ett spelkort. """
    VALORER = ["ess", "2", "3", "4", "5", "6", "7",
              "8", "9", "10", "knekt", "dam", "kung"]
    FARGER = ["Klöver", "Ruter", "Hjärter", "Spader"]

    def __init__(self, valor, farg, synligt = True):
        self.valor = valor
        self.farg = farg
        self.framsidanUpp = synligt

    def __str__(self):
        if self.framsidanUpp:
            rep = self.farg + "-" + self.valor
        else:
            rep = "XX"
        return rep

    def vand(self):
        self.framsidanUpp = not self.framsidanUpp
```

## HAND

- Klassen Hand representerar en korthand



- Klassens attribut är `korten` (en lista med kort).
- Metoderna är `__init__`, `__str__`, `bort` (tar bort korten), `stoppaIn` (lägger till ett kort), `ge` (ger ett kort till en annan korthand)

```
class Hand(object):
    def __init__(self):
        self.korten = []

    def __str__(self):
        if self.korten:
            rep = ""
            for card in self.korten:
                rep += str(card) + "\t"
            else:
                rep = "<tom>"
            return rep.ljust(14)

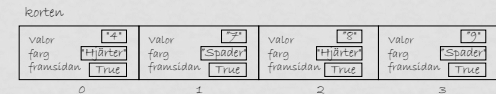
    def bort(self):
        self.korten = []

    def stoppaIn(self, card):
        self.korten.append(card)

    def ge(self, kort, annanHand):
        self.korten.remove(kort)
        annanHand.stoppaIn(kort)
```

## HAND-OBJEKT

- Ett Hand-objekt har bara ett attribut: `korten` som är en lista med Kort-objekt.
- Rita ett exempel:



## LEK

- Vi vill ha en klass `Lek` som representerar en hel kortlek.
- Vi skriver `class Lek (Hand)`
- Klassen `Lek` ärver då alla attribut och metoder från `Hand`.
- Vi definierar också tre nya metoder i klassen `Lek`: `fyll`, `blanda` och `delaUt`.

```
class Lek(Hand):
    """ En kortlek. """
    import random

    def fyll(self):
        for farg in Kort.FARGER:
            for valor in Kort.VALORER:
                self.stoppaIn(Kort(valor, farg))

    def blanda(self):
        random.shuffle(self.korten)

    def delaUt(self, personer, perHand = 1):
        for runda in range(perHand):
            for person in personer:
                if self.korten:
                    oversta = self.korten[0]
                    self.ge(oversta, person)
                else:
                    print("Slut på kort!")
```