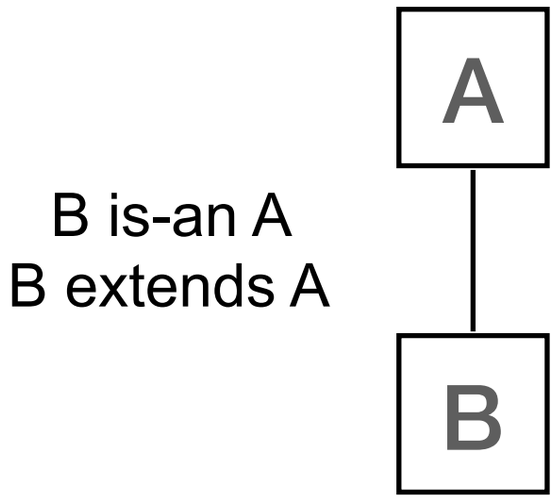
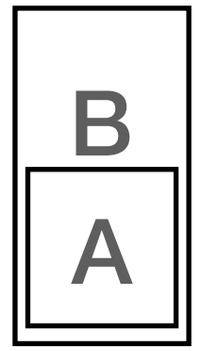
# Composition - UML

- Composition
  - an alternative to inheritance
- UML
  - Universal Markup Language
  - covers classes

# Composition

- Composition:
  - An alternative to inheritance to gain access to attributes and methods from another object




This might be a better way to extend built-in classes

# Remember: Extending the str class

- `intString` `__init__` calls the `str` `__init__`

```
class intString(str):
    def __init__(self,val):
        if (type(val) == int):
            str.__init__(val)
        else:
            print("Not a valid input")
    def to_int( self ):
            return int(self)
In [30]:
is2 = intString(34)
In [31]:
is3 = intString('sd')

Not a valid input
In [33]:
is2.isdigit()
```

# Using Composition

- intString has a `str` attribute
- Everything works except:
  - `i.isalpha()`
- Must use:
  - `i.s.isalpha()`
- Cannot directly access string methods

```
class intString():
    def __init__(self, s):
        if (type(s) == int):
            self.s = str(s)
        else:
            print("Not a valid input")
    def to_int(self):
            return int(self.s)

i = intString(34)

i.to_int()
Out[38]:
34

i.s.isalpha()
Out[41]:
False
```

# There is a method for that… `getattr()`

- You might want flexibility in deciding which attribute to get
  - attribute `name` not hardwired into the code

    `getattr(object, name)`

    is equivalent to:

    `object.name`

name can be decided elsewhere in the program

```
class attsClass():
    def __init__(self,aw,ax,ay,az):
        self.aw = aw
        self.ax = ax
        self.ay = ay
        self.az = az
```

```
ac = attsClass('W','X','Y','Z')
In [54]:
ac.ax
Out[54]:
'X'
In [55]:
getattr(ac,'ax')
Out[55]:
'X'
In [56]:
for att in ['X','Y','Z']:
    print(att)

X
Y
Z
```

# We need another method: `__getattr__()`

- `__getattr()__`
  - special method that will be called (if it has been defined) if an attribute access fails

```
class intString():
    def __init__(self, val):
        if (type(val) == int):
            self.s = str(val)
        else:
            print("Not a valid input")
    def to_int(self):
            return int(self.s)
    def __getattr__(self, attr):
        return getattr(self.s, attr)
```

```
In [43]:
i = intString(45)
In [44]:
i.to_int()
Out[44]:
45
In [45]:
i.isalpha()
Out[45]:
False
```

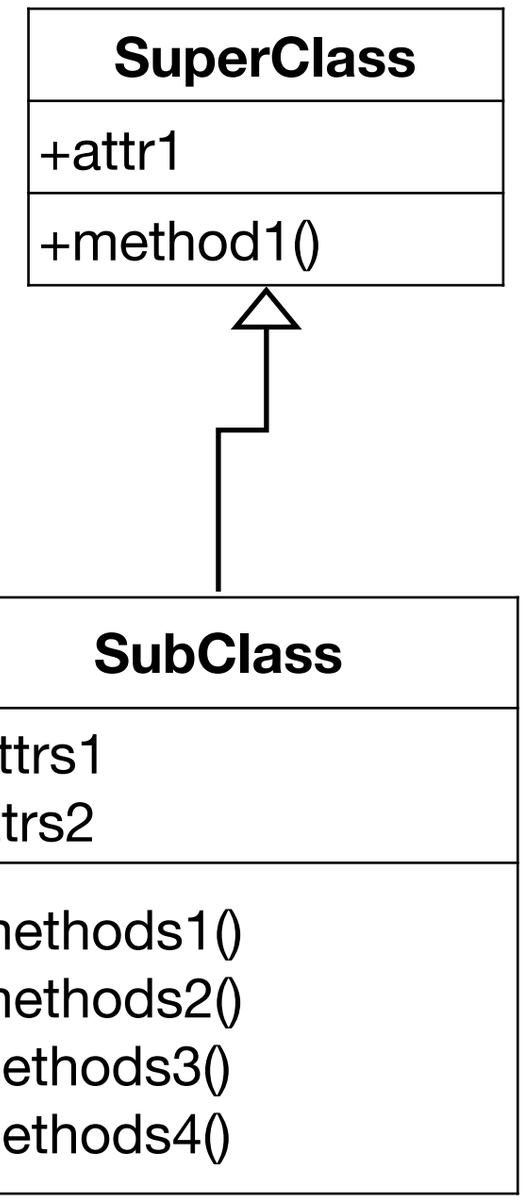
**Summary:**
Composition can be used (instead of inheritance) to get access to another class.
Use the `__getattr__()` / `getattr()` trick for direct access to the attributes of the other class.
If that is what you want

# UML class diagram

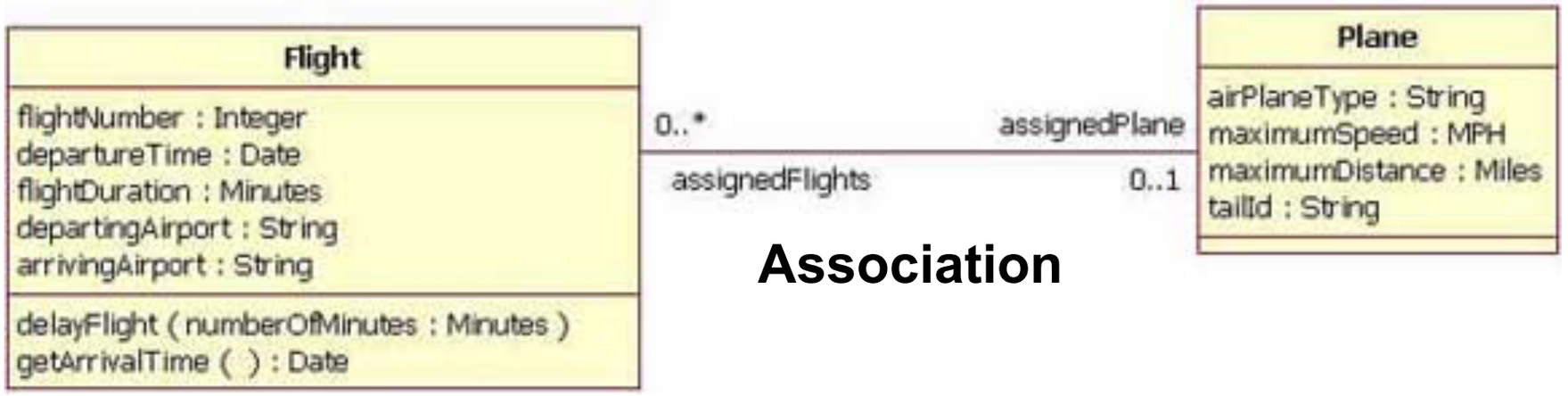
- Class name
- Box for attributes
- Box for methods
- + for public members
- ‘-‘ for private

- Inheritance
  - is-a relation
  - arrow (white triangle) at superclass

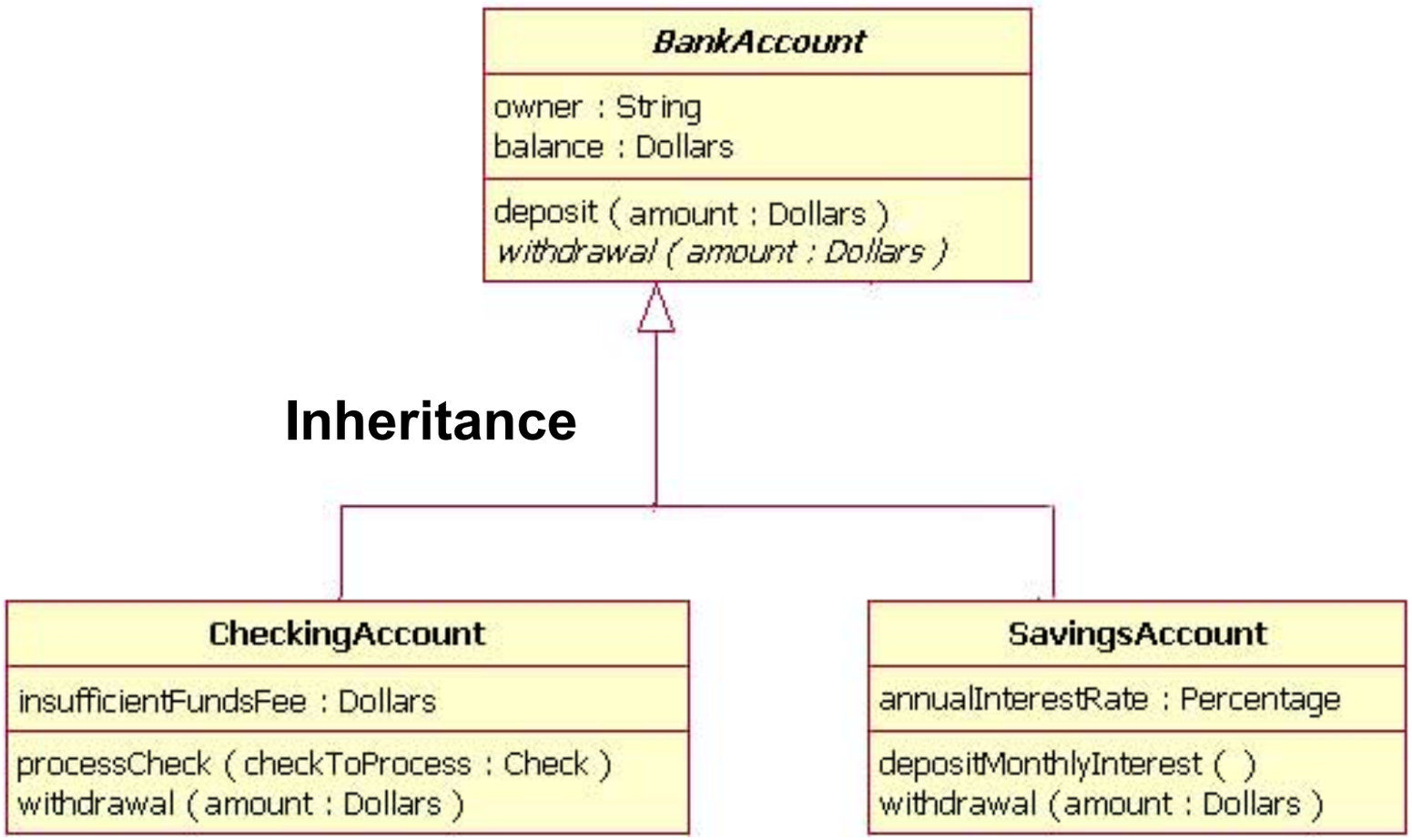
- Association
  - Cardinality
  - name



# UML Examples




**Association**




**Inheritance**

**Exercise**
Draw diagrams for
these associations:
AddressBook - ContactName
Mother - Child

https://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/index.html

# Exercise: Family Members

- Superclass Person
  - Person has attributes `name`, `surname` and `dob`, a constructor and a `show` method

```
pat = Person('Pat','Malone','18/03/2001')
pat.show()

Pat Malone  DOB: 18/03/2001
```

- Subclasses Parent and Child
  - Parent and Child are subclasses of Person
  - Parent has an additional attribute children - a list that is initially empty

# Exercise: Family Members

- Child has an additional attribute parents and a constructor that works like this:

```
joe = Parent('Joe','Bloggs','23/Dec/1982')
mary = Parent('Mary','Maher','19/Nov/1981')
lucy = Child('Lucy','Malone','11/Feb/2006',[])
joeJunior = Child('Joe Junior', 'Bloggs','14/Nov/2009',[joe,mary])
```

- The entries in the parents list must be parents. Otherwise error messages are produced.

```
luckyLaura = Child('Laura', 'White','14/May/2006',[joe,mary,lucy])
luke = Child('Luke','Smyth','19/Apr/2008',["Fido","Vlad"])

Lucy Malone is not a parent.
Fido isn't even a person
Vlad isn't even a person
```

- Child has a show method

```
joeJunior.show()

Joe Junior Bloggs  DOB: 14/Nov/2009
Parents
Joe Bloggs  DOB: 23/Dec/1982
Mary Maher  DOB: 19/Nov/1981
```

**Hint:** Call the show method of the parents.