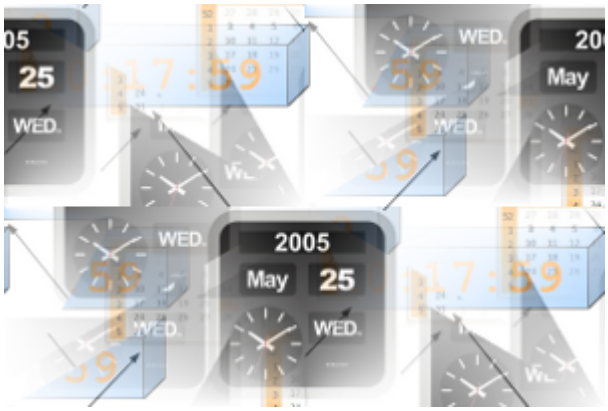
INHERITANCE EXAMPLE

INTRODUCTION



There aren't many good examples on inheritance available on the web. They are either extremely simple and artificial or they are way to complicated. We want to close the gap by providing an example which is on the one hand more realistic - but still not realistic - and on the other hand simple enough to see and understand the basic aspects of inheritance. In our previous chapter, we introduced inheritance formally.

To this purpose we define two base classes: One is an implementation of a clock and the other one of a calendar. Based on these two classes, we define a class CalendarClock, which inherits both from the class Calendar and from the class Clock.

THE CLOCK CLASS

```
class Clock(object):

    def __init__(self,hours=0, minutes=0, seconds=0):
        self.__hours = hours
        self.__minutes = minutes
        self.__seconds = seconds

    def set(self,hours, minutes, seconds=0):
        self.__hours = hours
        self.__minutes = minutes
        self.__seconds = seconds

    def tick(self):
        """ Time will be advanced by one second """
        if self.__seconds == 59:
            self.__seconds = 0
            if (self.__minutes == 59):
                self.__minutes = 0
                self.__hours = 0 if self.__hours==23  else
self.__hours
            else:
                self.__minutes += 1;
```

```
            else:
                self.__seconds += 1;

        def display(self):
            print("%d:%d:%d" % (self.__hours, self.__minutes,
    self.__seconds))

        def __str__(self):
            return "%2d:%2d:%2d" % (self.__hours, self.__minutes,
    self.__seconds)

    x = Clock()
    print(x)
    for i in range(10000):
        x.tick()
    print(x)
```

THE CALENDAR CLASS

```
class Calendar(object):
    months = (31,28,31,30,31,30,31,31,30,31,30,31)

    def __init__(self, day=1, month=1, year=1900):
        self.__day = day
        self.__month = month
        self.__year = year

    def leapyear(self,y):
        if y % 4:
           # not a leap year
           return 0;
        else:
           if y % 100:
               return 1;
           else:
               if y % 400:
                  return 0
               else:
                  return 1;

    def set(self, day, month, year):
        self.__day = day
        self.__month = month
        self.__year = year
```

```
    def get():
        return (self, self.__day, self.__month, self.__year)
    def advance(self):
        months = Calendar.months
        max_days = months[self.__month-1]
        if self.__month == 2:
            max_days += self.leapyear(self.__year)
        if self.__day == max_days:
            self.__day = 1
            if (self.__month == 12):
                self.__month = 1
                self.__year += 1
            else:
                self.__month += 1
        else:
            self.__day += 1


    def __str__(self):
       return str(self.__day)+"/"+ str(self.__month)+ "/"+
str(self.__year)

if __name__ == "__main__":
   x = Calendar()
   print(x)
   x.advance()
   print(x)
```

THE CALENDAR-CLOCK CLASS

```
from clock import Clock
from calendar import Calendar

class CalendarClock(Clock, Calendar):

   def __init__(self, day,month,year,hours=0,
minutes=0,seconds=0):
        Calendar.__init__(self, day, month, year)
        Clock.__init__(self, hours, minutes, seconds)

   def __str__(self):
       return Calendar.__str__(self) + ", " + Clock.__str__(self)
```

```
if __name__  == "__main__":
    x = CalendarClock(24,12,57)
    print(x)
    for i in range(1000):
        x.tick()
    for i in range(1000):
        x.advance()
    print(x)
```