

# 1 Object-oriented programming

*Object-oriented programming* is a programming paradigm that focusses on *objects* such as a person, place, thing, event, and concept relevant for the problem. Objects may contain data and code, which in the object-oriented paradigm are called *attributeds* and *methods*. Object-oriented programming is an extension of data types, in the sense that objects contains both data and functions in a similar manner as a module, but object-oriented programming emphasizes the semantic unity of the data and functions. Thus, objects are *models* of real world entities, and object-oriented programming leads to a particular style of programming analysis and design called *object-oriented analysis and design*.

- Object-oriented programming
- objects
- attributeds
- methods
- models
- object-oriented analysis
- object-oriented design

An object is a variable of a class type. A class is defined using the `type` keyword, and there are always parantheses after the class name. Consider the following problem.

## Problem 1.1

A complex number is a pair of real numbers called the real and the imaginary part and a set of operators. In particular, addition of two complex numbers is the the addition of their real parts and of their imaginary parts. Define a class for complex numbers including the addition operator.

A solution to this problem is as follows.

## Listing 1.1 complex.fsx:

A class implementing complex numbers and the addition operator.

```
1 type complex(aReal,anImaginary) =
2     member this.re = aReal
3     member this.im = anImaginary
4     member this.add (a : complex) : complex =
5         new complex(this.re + a.re, this.im + a.im)
6
7 let x = new complex(1.0,2.0)
8 let y = new complex(2.5,-1.2)
9 let z = x.add(y)
10 printfn "(%A, %A) + (%A, %A) = (%A, %A)" x.re x.im y.re y.im
    z.re z.im
```

---

```
1 $ fsharp --nologo complex.fsx && mono complex.exe
2 (1.0, 2.0) + (2.5, -1.2) = (3.5, 0.8)
```

Things to remember:

- upcast and downcast `upcast, ">:", downcast, "?:>"`
- boxing `(box 5) :?> int;;`, see Spec-4.0 chapter 18.2.6.

## *1 Object-oriented programming*

- obj type Spec-4.0 chapter 18.1
- boxing Spec-4.0 Section 18.2.6