CS 241 — Introduction to Problem Solving and Programming

Object-Oriented Programming

Third look at classes

Mar 18, 2005

Overview

An object is a unit that encapsulates data and functionality. A class is a specification for an object, containing instance variables and methods; these are members of the class.

An object fitting the class specification is an instance of the class. It is created by instantiating the class. It has its own copy of the instance variables, and when its methods are called, its own instance variables are used. When its methods are called, that object is the receiver of the method invocation.

```
x.m(a, b, c)
```

Example specification

Write a class for maintaining a set of readings (for example, of temperatures). Allow for new readings to be added to and removed from the set, and calculate the average, maximum, minimum, and range.

Invariants

A class invariant is a condition that holds for a class throughout its execution.

All methods (except the constructor) are to preserve the invariant. The constructor should set the invariant, that is, initialize the instance so that the invariant holds.

Constructors deal with initial conditions.

Accessibility

Members of a class can be given an accessibility level: `public` or `private`.

- An instance variable or method that is designated `public` can be accessed (read or written to; for instance variables) or invoked (for methods) by code in another class.

- An instance variable or method that is designated `private` can be be accessed (for instance variables) or invoked (for methods) only by code in the same class.

It is good programming practice to make all instance variables `private`.

Static

So what is static?

If a member of a class is `static`, then it belongs to a class, rather than an instance; all instances of the class share the same one.

```
public class Book {

    private static int currentId;
    private int uniqueId;

    Book() {
        uniqueId = currentId++;
    }
```

Static

If a method is `static`, it cannot refer to any instance variables or `this` or to non-`static` methods (except when specifying an object). Main methods are `static`.

The `Math` class has these `static` methods:

```
Math.pow(double, double)                              Math.abs(double)
math.round(double)                                    Math.sqrt(double)
math.floor(double)                                    Math.ceil(double)
```