

## 5 Abstracting over Data Definitions, Methods

### Portfolio Problems

Work out as complete programs the following exercises from the textbook. You need not work out all the methods, but make sure you stop only when you see that you really understand the design process.

#### Problems:

1. Return to the Problem 14.7 on page 144. Revise the class definitions and the method definitions to eliminate as much duplication as possible.
2. Return to the problem 3.1 from Assignment 3.  
Change the definition of the class `CartPt` so that it now extends the class `Posn` in the `geometry` package. Eliminate the method `toPosn` and run all tests again. Use this in your game, if appropriate.

### Pair Programming Assignment

#### 5.1 Problem

Complete Part 1 of the Lab 5, dealing with bank accounts.

Hand in (as two separate files) both the original solution and the solution after you have designed all abstractions.

#### 5.2 Problem

Complete one of the three problems that you have not done for the previous assignment. (If you have handed in problems 4.1 and 4.3, complete the problem 4.2, etc.)

#### 5.3 Problem

In this problem we examine the design of the classes for a bookstore - with print books, audio books and online books. The file `bookstore.java` contains the data definitions, examples of data, and the methods we have designed in class.

- A. Design the abstract class `ABook` that allows you to eliminate duplication of the field definitions. Make sure all your examples still work.
- B. The method `salePrice` becomes an abstract method with implementation in each of the subclasses. Complete this change and make sure the tests work correctly.
- C. Look at the method `sameAuthor`. See how you can design its implementation in the abstract class. Make sure you include the template and follow the design recipe. Run all tests.
- D. Add a re-designed method `writtenBy` to your classes.
- E. Design a new method `cheaperThan` that determines whether the sale price of one book is lower than the sale price of another book.
- F. Design a new method `titleBefore` that determines whether the title of one book is lexicographically before the title of another book.

## 5.4 Problem

### Creative Project

This week you will start the work on a new game. It may be a refinement of the game you have designed during the past four weeks.

*Note:* Hand in a written proposal for the game, similar to the first one. Make it look good - check the grammar, type it, add hand-drawn pictures. The proposal must follow these guidelines:

- A. The game must include be a collection of objects that change over time.
- B. The game must contain an object that changes its state in response to some interaction with the rest of the world, or in response to the time passed. (E.g. frog that is hit by a car, a rat that dies.)
- C. The proposal must include a *user document* that describes for the user how to play the game.
- D. The proposal must include a *design document* that describes the classes you will need to implement the game shown as a class diagram, and will describe the key methods you will need to implement. You only need to describe the purpose of the methods — possibly the information the methods will need.