Arash Lari & Mitchell Simon

Extra Credit Problems for 402

Problem 3.10, Rajlich page 46

Write a class that supports course grading; it contains an array of 30 students such that each student is identified by an integer and has a course grade. There is also a method that can change the student grade and a method that computes the grade point average for the entire class of 30 students.

graderFunction = studentArray => {

if (!studentArray || studentArray.length !== 30) {

throw "Only 30 students allowed."

}

this.students = studentArray;

}

graderFunction.prototype.changeGrade = function(student, grade) {

this.students[student] = grade;

}

graderFunction.prototype.gpaAverage = function() {

return this.students.reduce(function(a, b) {

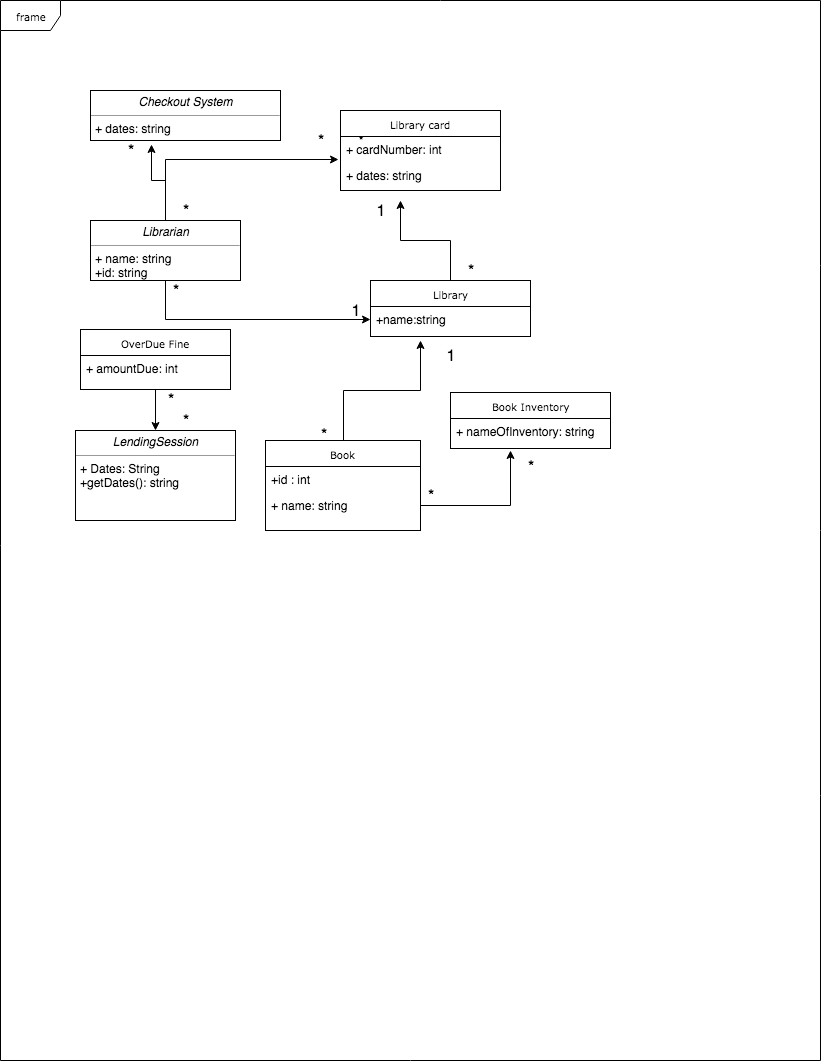
return a + b;

}) / this.students.length;

}

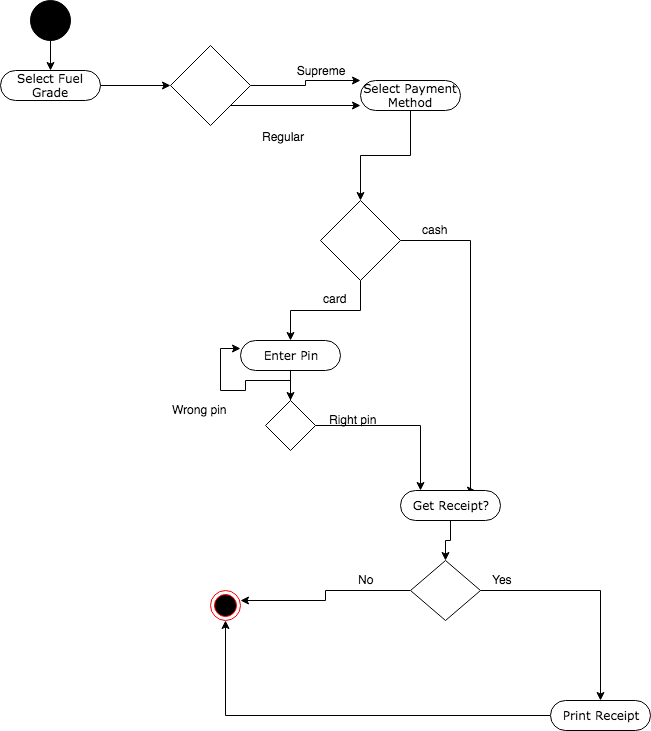
Problem 4.2, Rajlich page 63

Draw a class diagram of a library lending books using the following classes: Librarian, Lending Session, Overdue Fine, Book Inventory, Book, Library, Checkout System, and Library Card. Download and install a UML tool such as StarUML, ArgoUML, or Poseidon and use the tool to draw the class diagram.



Problem 4.3, Rajlich page 63

Draw an activity diagram of pumping gas and paying by credit card at the pump. Include at least five activities, such as "Select fuel grade" and at least two decisions, such as "Get receipt?" Download and install a UML tool such as StarUML, ArgoUML, or Poseidon and use the tool to draw the diagram. If you did this for the previous problem, you may use the same tool.



Problem 10.6, Schach page 166

For the following program design a minimal test suite:

public static void main(String args[]) { int i,j; int k = 0; i = Integer.parseInt(args[0]); j = Integer.parseInt(args[1]); if( i > 30 ) { if( i <= 60 && j <= 150 ) k = 1; else if( i <= 90 && j <= 150 ) k = 2; else k = 3; } if( i == j && i <= 30 ) k=4; System.out.println( k ); }

Test Suite:

public void testMain() {

assert( main(["50", "100") == "1" );

assert( main(["80", "100"]) == "2" );

assert( main(["100", "100"]) == "3" );

assert( main(["20", "20"]) == "4" );

}

Problem 10.8, Schach page 167

In test driven development, a test is written and then the code to pass it. Given the following test, write a method that passes the test.

public class TestGrade { Grade testGrade; public void testGetFinalGrade() { assert( testGrade.getFinalGrade(70) == "Pass" ); assert( testGrade.getFinalGrade(69) == "Fail" ); };

public String getFinalGrade(int grade) {

if (grade >= 70) {

return "Passed the test";

} else {

return "Failed the test";

}

}

Problem 10.8, Schach page 167

Expand the test in exercise 10.8 to include grades A to F using the following grade scale:

|  |  |
| --- | --- |
| **Letter Grade** | **Minimum Percentage** |
| A | 90 |
| B | 80 |
| C | 70 |
| D | 60 |
| F | Less than 60 |

public class TestGrade {

Grade testGrade;

public String getFinalGrade(int grade) {

if (grade >= 60) {

return "D"

} else if (grade >= 70) {

return "C"

} else if (grade >= 80) {

return "B"

} else if (grade >= 90) {

return "A"

} else {

return "F"

}

}

public void testGetFinalGrade() {

assert( testGrade.getFinalGrade(90) == "A" );

assert( testGrade.getFinalGrade(80) == "B" );

assert( testGrade.getFinalGrade(70) == "C" );

assert( testGrade.getFinalGrade(60) == "D" );

assert( testGrade.getFinalGrade(59) == "F" );

}

};