Brandon Kirklen
2015-05-10
CPE 102-22

Home Assignment 5

CPE 102

Chapter 8

1. What is a simple rule of thumb for finding classes?

Look for nouns.

2. Your job is to write a program that plays chess. Might ChessBoard be an appropriate class? How about MovePiece?

ChessBoard makes more sense because it's a noun. MovePiece would be a better name for a method rather than a class.

3. Why is the CashRegister class from Chapter 4 not cohesive?

Because it contains two distinct concepts, payments and coin tracking. This has multiple concepts and should be split.

4. Why does the Coin class not depend on the CashRegister class?

Because coin doesn't need any data from CashRegister to convert given coins.

5. Why is it a good idea to minimize dependencies between classes?

This would increase reusability.

6. Is the substring method of the String class an accessor or a mutator?

Accessor because the original string is not changed, in java strings are immutable.

7. Is the Rectangle class immutable?

No, moving the rectangle is mutation rather than accessor and recreation.

8. If a refers to a bank account, then the call a.deposit(100) modifies the bank account object. Is that a side effect?

Yes. However it's an expected side effect as the overall balance should change when more money is placed in the account.

9. Consider the Student class of Chapter 7. Suppose we add a method

Brandon Kirklen
2015-05-10
CPE 102-22

```
void read(Scanner in) {
        while (in.hasNextDouble()) {
                addScore(in.nextDouble());
        }
}
```

Does this method have a side effect other than mutating the scores?

It would take control of the "in" scanner and use a double values from it.

10. Suppose we want to count the number of transactions in a bank account in a statement period, and we add a counter to the BankAccount class:

```
public class BankAccount {
private int transactionCount;
. . .
}
```

In which methods does this counter need to be updated?

Deposit and Withdrawl methods are going to be adding to it. There also needs to be some sort of resetting of it. I think the better way to do this would be create a class which finds the number of transactions for a given period rather than attempting to keep a running total accurate. Each individual transaction would be needed anyway, so why complicate the class.

11. In How To 3.1, the CashRegister class does not have a getTotalPurchase method. Instead, you have to call receivePayment and then giveChange. Which recommendation of Section 8.2.4 does this design violate? What is a better alternative?

The giveChange method doesn't store the value returned. It just calculates it and then returns it without having a way to store it. A better concept would be to break this into a method which finds the difference between the amount paid and the amount owed and another method which returns the current amount owed. Then rather than having these functions clear the register, which is also a dumb name, the cash register isn't being recreated, the current ledger is what's being cleared but anyway, create a function which clears the current purchase and gets it ready for the next customer.

12. In the example in Section 8.3.3, why is the add method required? That is, why can't the user of a Question object just call the add method of the ArrayList<String> class?

Because the arraylist is private and can only be accessed within the class. This would be to prevent bad data from getting put in the list.

13. Suppose we want to enhance the CashRegister class in How To 3.1 to track the prices of all purchased items for printing a receipt. Which instance variable should you provide? Which methods should you modify?

Brandon Kirklen
2015-05-10
CPE 102-22

I would create a custom container class of which to make an arraylist out of which could store the string item name as well as the price, maybe a field for being taxable as well. You'd need to update the adding purchase method to use that variable as well as having a method which would print out the receipt at the end.

14. Consider an Employee class with properties for tax ID number and salary. Which of these properties should have only a getter method, and which should have getter and setter methods?

Salary should have both getter and setter methods. The answer the book is looking for is that the tax ID number shouldn't have a setter method because it will be static for the life of the employee class, but that's silly. Due to data entry errors there could be a need to change a tax ID number. This also is all based on how this particular class is implemented rather than universal ideals.

15. Suppose the setName method in Section 8.3.4 is changed so that it returns true if the new name is set, false if not. Is this a good idea?

Sure, it would be useful to know if a setter rejected the input.

16. Look at the direction instance variable in the bug example in Section 8.3.6. This is an example of which pattern?

A state variable

17. Name two static variables of the System class.

System.err and System.out

18. Name a static constant of the Math class.

Math.E

19. The following method computes the average of an array of numbers:

```
public static double average(double[] values)
```

Why should it not be defined as an instance method?

Because it doesn't require anything other than the array of values.

20. Harry tells you that he has found a great way to avoid those pesky objects: Put all code into a single class and declare all methods and variables static. Then main can call the other static methods, and all of them can access the static variables. Will Harry's plan work? Is it a good idea?

Sure but that's disgrossting programming practice. The whole point of objects and methods is to increase functionality, if you wanted to do all that, just go write assembly.

21. Which of the following are packages?

- java
- java.lang
- java.util
- java.lang.Math

java.lang and java.util are both packages. java.lang.Math is a class.

22. Is a Java program without import statements limited to using the default and java.lang packages?

No, you just have to fully qualify any method you want to use.

23. Suppose your homework assignments are located in the directory /home/me/cs101 (c:\Users\Me\cs101 on Windows). Your instructor tells you to place your homework into packages. In which directory do you place the class hw1.problem1.TicTacToeTester?

/home/me/cs101/hw1/problem1/TicTacToeTester

24. Provide a JUnit test class with one test case for the Earthquake class in Chapter 5.

```
public class EarthquakeTest {
        @Test public void testLevel1() {
                Earthquake quake = new Earthquake(1);
                Assert.assertEquals(
                "Most structures destroyed ",
                quake.getDescription());
        }
}
```

25. What is the significance of the EPSILON argument in the assertEquals method?

Because we're using floating points for these there's room for some small error between the values.

Chapter 9

1. Consider classes Manager and Employee. Which should be the superclass and which should be the subclass?

All managers are employees so employee is the superclass and manager is the subclass.

2. What are the inheritance relationships between classes BankAccount, CheckingAccount, and SavingsAccount?

BankAccount is the super class while checkingaccount and savingsaccount both come from the superclass.

Brandon Kirklen
2015-05-10
CPE 102-22

3. What are all the superclasses of the JFrame class? Consult the Java API documentation or Appendix D.

- java.awt.Frame
- java.awt.Window
- java.awt.Container
- java.awt.Component

4. Consider the method doSomething(Car c). List all vehicle classes from Figure 1 whose objects cannot be passed to this method.

Motorcycle, Truck, Vehicle,

5. Should a class Quiz inherit from the class Question? Why or why not?

No. A quiz has questions in it not a question has a quiz.

6. Suppose q is an object of the class Question and cq an object of the class ChoiceQuestion. Which of the following calls are legal?

- q.setAnswer(response)
  - legal
- cq.setAnswer(response)
  - legal
- q.addChoice(choice, true)
  - illegal, no addChoice method in Question
- cq.addChoice(choice, true)
  - legal

7. Suppose the class Employee is declared as follows:

```
public class Employee
{
private String name;
private double baseSalary;
public void setName(String newName) { . . . }
public void setBaseSalary(double newSalary) { . . . }
public String getName() { . . . }
public double getSalary() { . . . }
}
```

Declare a class Manager that inherits from the class Employee and adds an instance variable bonus for storing a salary bonus. Omit constructors and methods.

```
public class Manager extends Employee
{
```

```
double bonus;
//other stuff
}
```

8. Which instance variables does the Manager class from Self Check 7 have?

Name, base salary, and bonus

9. In the Manager class, provide the method header (but not the implementation) for a method that overrides the getSalary method from the class Employee.

```
@override
public double getSalary()
```

10. Which methods does the Manager class from Self Check 9 inherit?

getName, setName, setBaseSalary

11. What is wrong with the following implementation of the display method?

```
public class ChoiceQuestion {
        . . .
        public void display() {
        System.out.println(text);
                for (int i = 0; i < choices.size(); i++) {
                int choiceNumber = i + 1;
                System.out.println(choiceNumber + ": " + choices.get(i));
                }
        }
}
```

Text is a private variable and cannot be accessed.

12. What is wrong with the following implementation of the display method?

```
public class ChoiceQuestion {
        . . .
        public void display() {
                this.display();
                for (int i = 0; i < choices.size(); i++) {
                        int choiceNumber = i + 1;
                        System.out.println(choiceNumber + ": " + choices.get(i));
                }
        }
}
```

This.display is a recursive call.

Brandon Kirklen
2015-05-10
CPE 102-22

13. Look again at the implementation of the addChoice method that calls the setAnswer method of the superclass. Why don't you need to call super.setAnswer?

There is no setAnswer method in the subclass.

14. In the Manager class of Self Check 7, override the getName method so that managers have a * before their name (such as *Lin, Sally).

```
getName(){
        return "*" + super.getName;
}
```

15. In the Manager class of Self Check 9, override the getSalary method so that it returns the sum of the salary and the bonus.

```
getName(){
        return super.getSalary() + bonus;
}
```

16. Assuming SavingsAccount is a subclass of BankAccount, which of the following code fragments are valid in Java?

- BankAccount account = new SavingsAccount();
    - Valid
- SavingsAccount account2 = new BankAccount();
    - Invalid, mixing types
- BankAccount account = null;
    - Invalid
- SavingsAccount account2 = account;
    - Invalid assuming no existing savingsAccount "account"

17. If account is a variable of type BankAccount that holds a non-null reference, what do you know about the object to which account refers?

You know it's a part of the class BankAccount or one of it's subclasses.

18. Declare an array quiz that can hold a mixture of Question and ChoiceQuestion objects.

Question[] quiz = new Question[int x];

19. Consider the code fragment

```
ChoiceQuestion cq = . . .; // A non-null value
cq.display();
```

Which actual method is being called?

Unknown, it's of class ChoiceQuestion but a subclass could override the display method.

Brandon Kirklen
2015-05-10
CPE 102-22

20. Is the method call Math.sqrt(2) resolved through dynamic method lookup?

No, the sqrt function is a part of the Math class.

21. Why does the call

```
System.out.println(System.out);
```

produce a result such as java.io.PrintStream@7a84e4?

Because there is no “toString” method in the implementation

22. Will the following code fragment compile? Will it run? If not, what error is reported?

```
Object obj = "Hello";
System.out.println(obj.length());
```

No, obj has no method length.

23. Will the following code fragment compile? Will it run? If not, what error is reported?

```
Object obj = "Who was the inventor of Java?";
Question q = (Question) obj;
q.display();
```

This will compile because you’re force casting it to a question type. The first line compiles trivially because object has a string subtype. The final line will also compile because q is of question type and question has a display method. However there will be a runtime error due to the force cast.

24. Why don’t we simply store all objects in variables of type Object?

Because type object has relatively few methods you can call on it because typing is a good idea.

25. Assuming that x is an object reference, what is the value of x instanceof Object?

False if x is null and true if x is a value.