CS161: Introduction to Computer Science
Homework Assignment 4
Due: 2/15 by 11:59pm

So far you have seen three different classes: a `Die` class, a `BankAccount` class, and a `Student` class. Variations of the `Die` and `BankAccount` classes are also in the reading in chapter 4. This assignment asks you to write another class[1]!

ORCA Card

This assignment asks you to write a Java class that simulates an ORCA card. An ORCA card is used to pay for bus, train, and ferry trips in Pierce and King counties. When using the real card, you add funds to it and can then swipe the card to charge rides on the various services. In addition to keeping track of the balance and paying for rides, "our" ORCA card will also keep track of the number of trips that have been taken.

For full credit, your class should contain all of the methods described below. They should have exactly the same name as shown, take the correct arguments, and return the correct information. (I will run a program that creates instances of your class and tests them, and if your names or other details differ, my testing code won't compile.) The assignment is less specific about the instance variables you'll need – you'll have to figure out what you need to store in order to implement the methods below.

Programming Questions

1. Create a new BlueJ project named `hw4`. Inside, create a Java class called `OrcaCard`.

2. Your class needs to be able to keep track of how much money is currently stored on the card. This balance should be set by the constructor, which should take a single argument (a double) specifying the initial balance.

3. Your class should have a `topUp` method that takes a single argument (the amount to add to the current balance) and adjusts the balance but doesn't return anything.

4. We'll simulate the process of "swiping" the card via the `buyTrip` method. This method should take a double (representing the cost of the trip). Inside the method, you need to decrease the balance by the cost of the ticket *plus the cost of taxes.*

For example, in Washington state, taxes are 6.5%. Let's say I want to take a ferry from Tacoma to Vashon Island. The cost of the ticket itself is $5.25. So, if I call your method and pass in `5.25`, the balance on my card should decrease by $5.25 plus an additional 0.34 cents for taxes.

Use a 6.5% tax rate in your code. A good idea is to store this value in an instance variable.

5. We'll also add a `getAverageTripCost` method. It doesn't need any arguments, but should return the average cost of the trips paid for by this card so far.

6. Finally, write a toString method that returns a string representation of the ORCA card object. Feel free to personalize this as you see fit, but the string should contain at least the card's current balance and the number of trips taken.

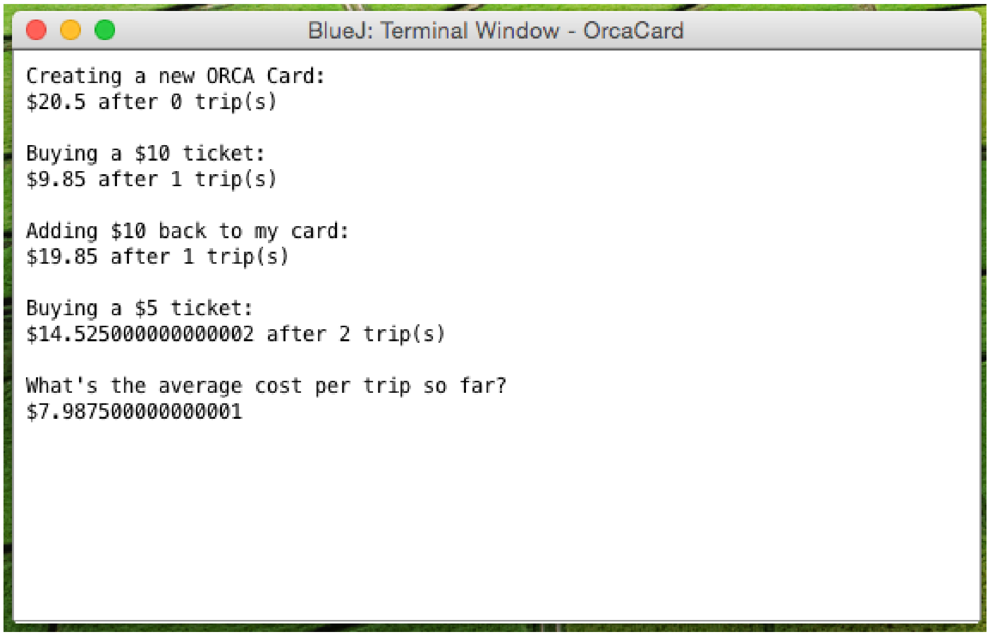
7. For full credit, your code should contain comments. There should be a block comment at the top of the class containing your name and a sentence or two describing the class. In addition, add a block comment before each method describing what the method does. Finally, use inline comments (//) for any code that you think is complicated and should be explained further.

[1] If you haven't noticed yet, the skill we're working on right now is how to write Java classes!

Here's an example of how I might use your `OrcaCard` class:

```
public static void main(String[] args){

    // Create a new card with a balance of $20.50
    System.out.println("Creating a new ORCA Card:");
    OrcaCard oc = new OrcaCard(20.50);
    System.out.println(oc.toString());

    System.out.println("\nBuying a $10 ticket: ");
    oc.buyTrip(10.00);
    System.out.println(oc.toString());

    System.out.println("\nAdding $10 back to my card: ");
    oc.topUp(10.00);
    System.out.println(oc.toString());

    System.out.println("\nBuying a $5 ticket: ");
    oc.buyTrip(5.00);
    System.out.println(oc.toString());

    System.out.println("\nWhat's the average cost per trip so far?");
    double avg = oc.getAverageTripCost();
    System.out.println("$" + avg);
   }
```

Here is what would be printed to the screen:



```
Creating a new ORCA Card:
$20.5 after 0 trip(s)

Buying a $10 ticket:
$9.85 after 1 trip(s)

Adding $10 back to my card:
$19.85 after 1 trip(s)

Buying a $5 ticket:
$14.525000000000002 after 2 trip(s)

What's the average cost per trip so far?
$7.987500000000001
```

Note that we purchased two tickets for $10 and $5 dollars respectively. If we don't take taxes into consideration, the average trip cost is $7.50 dollars per trip. However, if we take taxes into consideration, we're averaging about $7.99 dollars per trip.

Extras

Looking for something more to do?! How about adding code to your `buyTrip` method so that it also prints out a simulated ticket that shows (for example): the cost of the ticket, the amount paid in taxes, the remaining balance on the card, and a random time of day. This would be very useful so that we don't have to call the `toString` method after buying a ticket to see our remaining balance.

Submitting your assignment

You should submit your `hw4` folder with your `OrcaCard` class inside. Please remember to rename your folder

```
hw4_firstName_lastName
```

before you zip it.