CSE 1322L - Lab 4

Introduction

In this lab, you will write a program that allows guests at a theme park to keep track of how many tickets they have won while at the park, as well as allow guests to exchange said tickets for prizes. Our program will keep track of only one wallet, but we'll structure things in such a way that would allow us to expand the number of wallets being used simultaneously.

Note that, going forward, whenever you are asked to write a class, unless stated otherwise, **class fields should always be private while class methods and constructors should always be public**.

Methods besides getters and setters will usually be described according to their intended header:

unexpected_modifiers return_type name(type, type, …)

for example:

static double cos(double)

The method above should be read as being static, returning a double, with the name "cos", and taking one argument of type double. Methods with no modifiers should be simply set to "public". Lastly, while you are free to name your parameters anything you like, you are encouraged to name them sensibly. For the method above, a good name would be "angle" (as in "static double cos(double angle)"), as the method will take in an angle and return its cosine. The type and order of the parameters must still match the header described. The same applies to constructors.

Requirements

The features described below must be in your program:

- A class called ParkWallet
  - It must have 2 fields, one called "tickets", which is an integer, and one called "holiday", which is a static boolean initialized with "false".
    - The first field is to keep track of how many tickets the guest has, is to keep track of if it is currently a holiday or not, as prices are half off during holidays.
  - **ParkWallet()**: Sets tickets to 0
  - **ParkWallet(int)**: Sets tickets to the passed argument
  - Both fields must have getters

    - Note that, since holiday is static, its getter must also be static
  - Both fields must have setters
    - The setter for tickets should ignore any values which would set tickets to a negative value
    - The setter for holiday should not take any arguments but, instead, simply flip the current value in holiday (from true to false or vice versa).
    - Note that, since holiday is static, its setter must also be static
  - **void addTickets(int)**: If the argument is non-negative, increment tickets by the argument (e.g.: if a wallet has 5 tickets, addTickets(5) would bring the wallet up to 10 tickets). Otherwise, do nothing.
  - **boolean removeTickets(int)**: If the argument is non-negative and there are enough tickets in the wallet, decrement that many tickets from the wallet and return true (e.g.: if a wallet has 10 tickets, removeTickets(2) would bring the wallet down to 8 tickets). Otherwise, return false.
- Driver
  - Create a ParkWallet with an initial 100 tickets
  - In a loop, implement the following menu options:
    1. **Add tickets**: Prompts the user for the number of tickets to add to the wallet, incrementing the wallet with the number of tickets entered.
    2. **Set tickets**: Prompts the user for the number of tickets to which their wallet should be set to.
    3. **Buy prize**: Shows the user 3 prizes they can buy: a t-shirt, a sun hat, and a pair of sneakers. The prizes are worth 150, 350, and 600 points respectively. If the user has enough tickets to buy a prize, deduce that many tickets from their wallet and then print a saying the purchase went through, otherwise, print an error message.
       - **If it is currently a holiday, the above prices should be 50% off**
    4. **Set holiday**: Flips the ParkWallet static field to its current opposite, using the appropriate method
    5. **Quit**: Terminates the program

Deliverables

Lab4.java

ParkWallet.java

Considerations

- Despite what the deliverables say above, you can place all of your classes in a single file
  - In general, you will want to place different classes on different files. This helps organize your project.

Sample Output (user input in red)

```
[Theme Park Wallet Manager]
1. Add tickets
2. Set tickets
3. Buy prize
4. Set holiday
5. Quit
Your wallet has 100 tickets
Enter option: 1

Add how many tickets? 50

1. Add tickets
2. Set tickets
3. Buy prize
4. Set holiday
5. Quit
Your wallet has 150 tickets
Enter option: 2

Set ticket balance to: 300

1. Add tickets
2. Set tickets
3. Buy prize
4. Set holiday
5. Quit
Your wallet has 300 tickets
Enter option: 3

1. T-shirt (150 tickets)
2. Sun hat (350 tickets)
3. Sneakers (600 tickets)
Buy which prize? 2
Not enough tickets to buy a sun hat

1. Add tickets
2. Set tickets
3. Buy prize
4. Set holiday
5. Quit
Your wallet has 300 tickets
```

```
Enter option: 4

It is now a holiday.

1. Add tickets
2. Set tickets
3. Buy prize at holiday prices!
4. Set holiday
5. Quit
Your wallet has 300 tickets
Enter option: 3

Holiday Prices!
1. T-shirt (75 tickets)
2. Sun hat (175 tickets)
3. Sneakers (300 tickets)
Buy which prize? 2
Bought a sun hat for 175 tickets

1. Add tickets
2. Set tickets
3. Buy prize at holiday prices!
4. Set holiday
5. Quit
Your wallet has 125 tickets
Enter option: 4

It is no longer a holiday.

1. Add tickets
2. Set tickets
3. Buy prize
4. Set holiday
5. Quit
Your wallet has 125 tickets
Enter option: 5

Shutting off...
```