CS111 Recitation 8

Exercise 1: Arrays and Strings

Write a program that will take a string and separate the characters in the string into an array. So if the user enters “Puppies,” create an array of length 3 with “P” at index 0, “u” at index 1, and so on. With this array, reverse the string and print it.

Exercise 2: Objects

Create a class Employee that will represent a worker in some unknown industry. Each employee should have a name, bank account balance and an hourly rate that is unique to them. So Worker1 can have a rate of $5 per hour and Worker2 can have $8 per hour. Write 3 methods for the employee class – a “work(int hours)” method that adds money to that worker's balance, a “spend (double money)” method, and a “printinfo()” method which prints that workers name, rate, and balance. Add 2 constructors – one that lets you specify a starting balance and the rate, and another that only lets you specify the rate and defaults the balance to 0.

Now write a program that creates 3 workers, assigns them to work a few shifts, spend some of their money, then prints out the information for each worker.

As an extra challenge, you can prompt the user to create a “custom” Employee by asking for name, starting rate, etc, and storing these in an array.

Example run:

```
java Employee
creating employees
Employee name is Bob
Employee rate is 50
Employee balance is 3000

Employee name is Rob
Employee rate is 20
Employee balance is 1000

Employee name is Tom
Employee rate is 10
Employee balance is 0

working employees a few hours
Employee name is Bob
```

```
Employee rate is 50
Employee balance is 3500

Employee name is Rob
Employee rate is 20
Employee balance is 1400

Employee name is Tom
Employee rate is 10
Employee balance is 300
All done, going home!
```

Exercise 3: Arrays

For this exercise you will write a very simple board game called “nomopoly”. In this game, 4 players compete to see who can run out of money first. The rules are as follows:

- The game plays itself – your game loop will roll a 3 sided dice to move the 4 players. The player to get to 0$ first wins.
- The board will be represented by an array of length 10. When a player gets past the end, move them back to the start.
- Each square has a price which is the amount of money the player loses for landing on it.

Optional and significantly more difficult:

- Each square should have an owner and a price, not just a price. Assign prices however you -want. They can all be the same, or a multiple of the square's position in the board, anything works.
- The first player ever to land on a square pays the price and becomes the owner.
- All subsequent players landing on that same square pay the owner the price of that square.

Here is an example run that contains the optional component:

```
java Nomop
Created new player with $1000 at square 0
Created new player with $1000 at square 0
Created new player with $1000 at square 0
Created new player with $1000 at square 0
Created new Square with price200
Created new Square with price300
...[more of this] ...
Created new Square with price100
Starting game:
```

```
=======================================
rolled a 3 for player 0
Player 0 is now on 3
Player 0 now owns block 3 and has $700
 =======================================
rolled a 3 for player 1
Player 1 is now on 3
Player 1 now just paid 300 to player 0 and has $700
=======================================
...[turns skipped] …
=======================================
rolled a 1 for player 3
Player 3 is now on 2
Player 3 now just paid 200 to player 3 and has $600
=======================================
rolled a 3 for player 0
Player 0 is now on 4
Player 0 now just paid 100 to player 3 and has $1200
=======================================
rolled a 2 for player 1
Player 1 is now on 7
Player 1 now just paid 100 to player 0 and has $200
=======================================
rolled a 2 for player 2
Player 2 is now on 8
Player 2 now just paid 100 to player 1 and has $0
Player 2 has run out of money and won!

Player wallets and positions:
Player 0 is at 4 with $1300
Player 1 is at 7 with $300
Player 2 is at 8 with $0
Player 3 is at 2 with $700
```