CombinationLock

Directions

In this assignment you are going to write a program that represents the functionality of a combination lock.

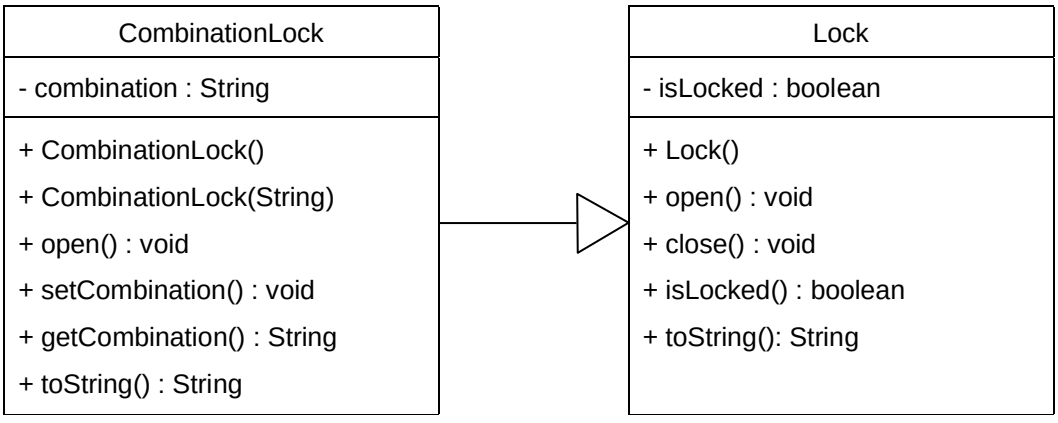
Design

There are many types of locks including combination locks, pad locks, and dead bolt locks, just to name a few. What do all these locks have in common? Answer - they are either in the state of locked or unlocked. Therefore we can design a class that represents the behavior that all locks possess, hence a more general description. A combination lock is a type of lock but it is more specific in how it state changes; namely entering a combination. Therefore we will design a second class that represents the behavior of a combination lock and its more specific description. Through inheritance the combination lock class will inherit the behavior of the general lock class.

UML Class Diagram

UML (Unified Modeling Language) is a pictorial language that allows you to describe a software system in visual way. A **class diagram** is a type of UML diagram that describes the attributes and operations of a class. A class diagram also shows the relationship between classes within a software system and it shows the responsibility each class has within the system.

Below is a class diagram for the CombinationLock class that extends the Lock class.



Here is a link to a tutorial on UML.

Lock Class

Lock Class - Instance Variables

Here is the starter code for the Lock class:

```
public class Lock
{
    //instance variables

    //constructor, starts off locked
    public Lock(){

    }

    //setter methods;
    public void open() {    }
    public void close() {    }

    //getter methods
    public boolean isLocked() {    }

    //toString method
    public String toString(){

    }
}
```

Lock Class – Instance Variables and Methods

- Instance Variables
    - private boolean isLocked;
- Constructor
    - No parameters
    - Initializes isLocked to true
- Setter Methods
    - Sets isLocked to appropriate value
- Getter Method
    - Returns the boolean value of isLocked
- toString()
    - returns either "Lock is open." or "Lock is closed."

CombinationLock Class

The Lock class is now our base or super class. The next step is to derive a class named CombinationLock from our base class. This class will inherit the behavior of the Lock class but also extend its behavior to include a combination.

Here is the starter code

```
public class CombinationLock extends Lock
{
    //instance variables

    //constructors
    public CombinationLock(String combination){    }
    public CombinationLock(){    }

    //setter methods
    @Override
    public void open(){    }
    public void setCombination(){    }

    //getter method
    public String getCombination(){    }

    @Override
    public String toString(){    }
}
```

CombinationLock Class – Instance Variables and Methods

- Instance Variables
    - private String combination;
- Constructors
    - Both need to call the Lock() constructor and initialize the combination.
    - The default constructor sets the combination to "" while the other to the String parameter.
- Setter Methods
    - No need to override the lock() method since locking doesn't require anything new for a combination lock
    - Since a combination is unique type of lock we will override the open() method. It should ask the user to enter the combination. If it's correct, then open the lock and say it's open. If incorrect, state that their input is wrong. Use Lock's toString() method to say it's open or not.

    - setCombination() If the lock is open, this method should ask the user to enter a new combination. If it's closed, tell the user "Unlock First".
    - getCombination() If the lock is closed, tell the user to "Unlock First". If it's open, go ahead and return toString().
- toString()
    - Since Lock has a toString(), then this one will override it. Use Lock's toString() to state whether the lock is open or closed. Then print out what the combination is in a pleasing format.

Test Class

We have finished the implementation of the CombinationLock class. The next step is to test our class to make sure it works the way we intended. To do this we are going to create a third class named CombinationLockTest that is a client of the CombinationLock class.

CombinationLockTest code

Copy the code below into a source file named **CombinationLockTest.java.**

```
import java.util.Scanner;
public class CombinationLockTest
{
    public static void main(String[] args)
    {
        Scanner inKey = new Scanner(System.in);
        CombinationLock myLock = new CombinationLock("11-22-33");
        //menu
        int choice = 0;
        while (choice != 6){
            System.out.println("\n   Menu");
            System.out.println("--------------------------");
            System.out.println("  1 - Check Lock");
            System.out.println("  2 - Unlock");
            System.out.println("  3 - Lock");
            System.out.println("  4 - Check Combination");
            System.out.println("  5 - Change Combination");
            System.out.println("  6 - Quit");
            System.out.print("Enter Choice: ");
            choice = inKey.nextInt();
            System.out.println();
            switch(choice){
                case 1:
```

```
                    if (myLock.isLocked()) System.out.println("Lock is closed
.");
                    else System.out.println("Lock is open.");
                    break;
                case 2: myLock.open(); break;
                case 3: myLock.close(); System.out.println("Lock is Closed");
break;
                case 5: myLock.setCombination(); break;
                case 4: System.out.println(myLock.getCombination()); break;
                case 6: break;
                default: System.out.println("Not a recognized selection");
            }
        }
    }
}
```

Now start testing!!

Sample Run

The functionality should match, not necessarily the verbiage. User input is bolded.

```
      Menu
   --------------------------
     1 - Check Lock
     2 - Unlock
     3 - Lock
     4 - Check Combination
     5 - Change Combination
     6 - Quit
   Enter Choice: 1

   Lock is closed.

      Menu
   --------------------------
     1 - Check Lock
     2 - Unlock
     3 - Lock
     4 - Check Combination
     5 - Change Combination
     6 - Quit
   Enter Choice: 4

   Lock is closed.
   Please open before changing combination.

      Menu
   --------------------------
     1 - Check Lock
     2 - Unlock
     3 - Lock
```

```
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 5

Lock is closed.
Please open before changing combination.

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 2

Enter the combination: 123
Combination does not match.
Lock is closed.

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 2

Enter the combination: 11-22-33
Lock is open.

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 4

Lock is open.
Combination = 11-22-33

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 5
```

```
Enter the new combination: hello there
Combination has been changed.

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 1

Lock is open.

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 3

Lock is Closed

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 1

Lock is closed.

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 2

Enter the combination: hello there
Lock is open.

   Menu
--------------------------
  1 - Check Lock
  2 - Unlock
```

```
  3 - Lock
  4 - Check Combination
  5 - Change Combination
  6 - Quit
Enter Choice: 6
```