

JC2002 Java Programming - Practical 3 (Day 3)

The goal of this practical is to familiarise you with object oriented programming concepts, especially class inheritance and method overriding, and to get some practice in defining and using classes and methods in Java programs.

1. Define class **Door** in file '**TestDoor.java**' with one protected instance variable **isOpen** of boolean type, initialised as false. In addition, define two public methods, **openDoor()** and **closeDoor()**, with the following functionality:
 - When **openDoor()** is invoked and **isOpen** is false, print text "Door opened", set **isOpen** to true, and return.
 - When **openDoor()** is invoked and **isOpen** is true, print text "Door already open!" and return.
 - When **closeDoor()** is invoked and **isOpen** is false, print text "Door already closed!" and return.
 - When **closeDoor()** is invoked and **isOpen** is true, print text "Door closed", set **isOpen** to false, and return.
2. Define public class **TestDoor** with **main()** method in the same file '**TestDoor.java**' that creates an instance of **Door** and loops for user feedback. Inside the loop, you can read user input as integers with the following options:
 - 1: Open door
 - 2: Close door
 - 3: Exit

You can use **switch...case** structure to implement different functionalities for the menu options. In case user enters 1, invoke **openDoor()** method of the **Door** instance. In case user enters 2, invoke **closeDoor()** method of the **Door** instance. In case user enters 3, exit the loop and end the program. In case user enters something else (default), print "Invalid option!" and continue the loop.

Test the program by using different options when the door is closed and open to ensure it works correctly.

3. Define class **DoorWithLock** that extends (inherits) class **Door**, including new boolean type instance variable **isLocked** initialised as false. You can still use the same file '**TestDoor.java**'. Override method **openDoor()** so that if **isOpen** is false and **isLocked** is true, the method prints text "Door is locked and cannot be opened!" and returns. Otherwise, the functionality of **openDoor()** should be the same as in class **Door**.

In addition, implement two new methods **lockDoor()** and **unlockDoor()** with the following functionality:

- When **lockDoor()** is invoked and **isOpen** is true, print text “Open door cannot be locked!” and return.
 - When **lockDoor()** is invoked and **isOpen** is false and **isLocked** is true, print text “Door already locked!” and return.
 - When **lockDoor()** is invoked and **isOpen** is false and **isLocked** is false, print text “Door locked”, change **isLocked** to true, and return.
 - When **unlockDoor()** is invoked and **isLocked** is false, print text “Door is not locked!” and return.
 - When **unlockDoor()** is invoked and **isLocked** is true, print text “Door unlocked”, change **isLocked** to false, and return.
4. Modify method **main()** in **TestDoor** class to test class **DoorWithLock** with additional menu options for locking and unlocking the door (i.e., invoking **lockDoor()** and **unlockDoor()** methods), in addition to opening and closing. The menu options could look as follows:
- 1: Open door
 - 2: Close door
 - 3: Lock door
 - 4: Unlock door
 - 5: Exit

Run the program and test it with different cases, such as trying to lock the door when the door is open, to make sure that the methods in class **DoorWithLock** are working properly.

5. Define class **DoorWithCodeLock** that extends (inherits) class **DoorWithLock**, including new int type instance variable **code** initialised as 0.
- Override method **lockDoor()** so that when the door is closed and unlocked, the method takes an integer input from the keyboard and saves the value in instance variable **code** before it changes **isLocked** to true and returns. In addition, override method **unlockDoor()** so that in case the door is locked, it takes an integer input from the keyboard and compares it to the value of **code**. If the user input matches **code**, the door is unlocked, but if the user input does not match **code**, text “Invalid code!” is printed and the door remains locked.
6. Modify **main()** method of **TestDoor** to instantiate **DoorWithCodeLock** instead of **DoorWithLock**, and run the program to test that the code lock works properly.