PROJECT REPORT

PREPARED BY: EGE ERGÜN

PROJECT REPORT

Our project is about connecting in internet site you must login however second login , login not needed because while you connecting on session you have token with this token internet site knows you. My program stores user info’s on different memory devices second login it identify the user and login not needed storing because we store the user data inside the memory devices. This project using a lot of Object Oriented Programming concepts hence there are used Composition - Inheritance - Polymorphism - Interfaces - Exceptions -Singleton pattern - Prototype pattern-Adapter pattern-Iterator pattern-Null Object Pattern- Interfaces. My main class computer. Computer has a memory device I am using composition. USB stick and Smartcard classes inherit from memory device and using memory device using methods.

METHODS:

Smart-Card

In order to be able to write something to the smart card the following steps are performed.

1.wait for card insertion 2. verifyPIN 3. select a file 4. encrypt data 5. write data to the card

In order to be able to read from the smart card the following steps are performed.

1. wait for card insertion 2. verifyPIN 3. select file 4. read data 5. decrypt data

USB Stick

In order to be able to write something to the USB stick the following steps are performed.

1. wait for USB stick Insertion 2. open file 3. encrypt data 4. write data to the card 5. close file.

In order to be able to read from the USB stick the following steps are performed.

1. wait for USB stick insertion 2. open file 3. read data 4. close file 5. decrypt data

Software Patterns

Adapter

I used adapter pattern because our application have 2 incompatible devices that uses the same methods. My project’s goal is to write an application that will support both a smart card and a USB flash. For this purpose I created an abstract class named USBMemoryDevice which contained the basis methods for my project like decryptData() and encryptData(). Then I used adapter class for adapting the USBMemoryDevice class to FlashUSB and SmartCardUSB classes.

Iterator

My program have methods that has to be repetitive over time. These methods needs to be reading the data or deleting the files and these methods needs to be done in a repetitive matter. Iterator pattern is looking like the most rational pattern to implement here because of the simplicity and reliability of the pattern and our need for making these methods iterative without exposing the program.

Singleton

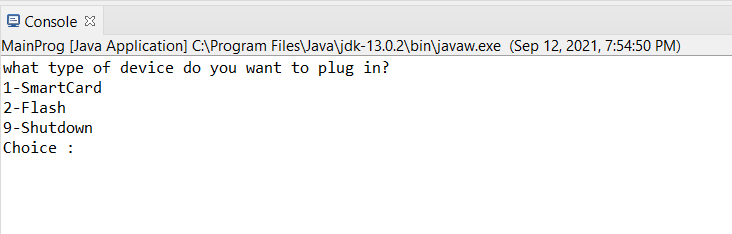
The singleton design pattern is used to restrict the instantiation of a class and ensures that only one instance of the class exists in the JVM. In other words, a singleton class is a class that can have only one object (an instance of the class) at a time per JVM instance. I used singleton pattern in ComputerSingleton class, for this intention, in our example, only one instance of computer can be created.

Null Object Pattern

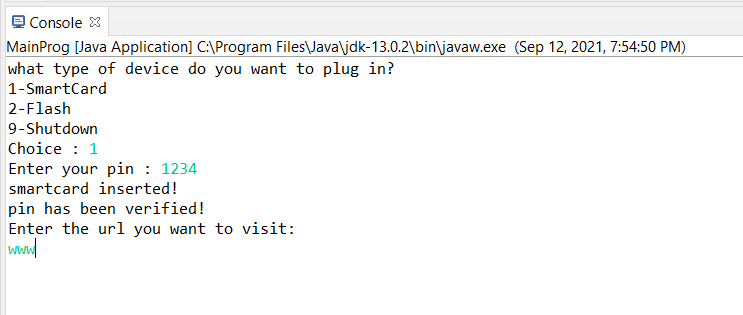
In Null Object pattern, a null object replaces check of NULL object instance. Instead of putting if check for a null value, Null Object reflects a do nothing relationship. Such Null object can also be used to provide default behaviour in case data is not available.

Prototype

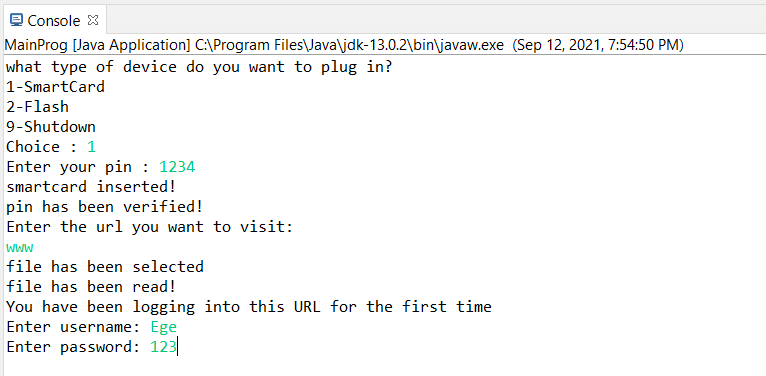
I create SmartCard and FlashUSB objects and store them in hash table. We use of copies of these objects when we need them. We create objects with load cache then we fetch object from repository actually we fetch a copy of created object. Because we don’t want to disturb the template in the hash table. USB memory device has clone, other adapter classes inherit from USB memory device. When we create one factory adapter we call createAdapter() which is it’s factory adapter we call back a smart card object. The program allows us to return a copy of the smart card object to one of those in memory. It gives from the cache the ready cached copy.

Picture 1(Storing user info smartcard device)

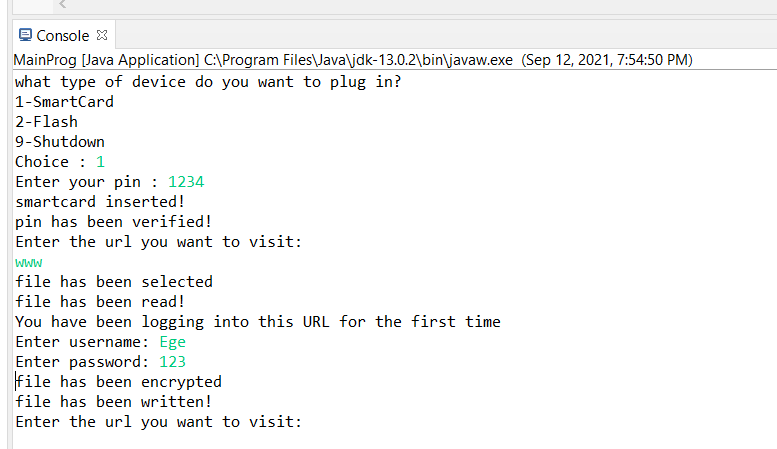
Picture 2



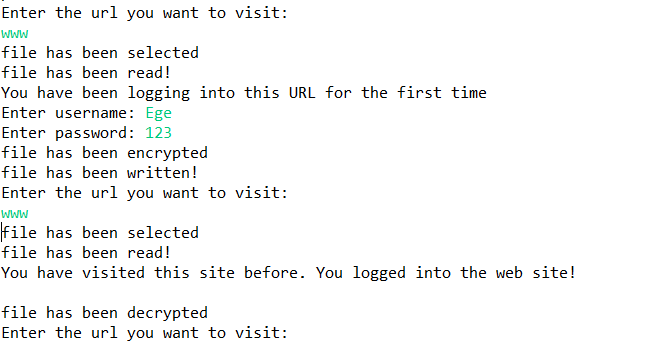
Picture 3



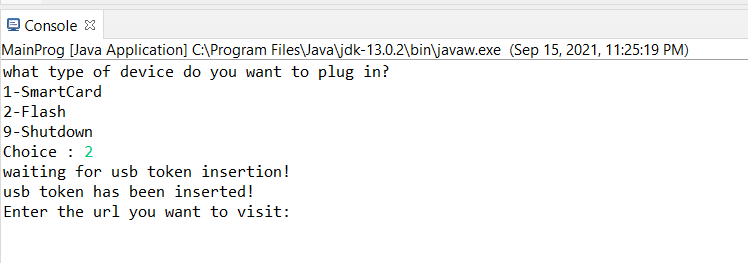
Picture 4



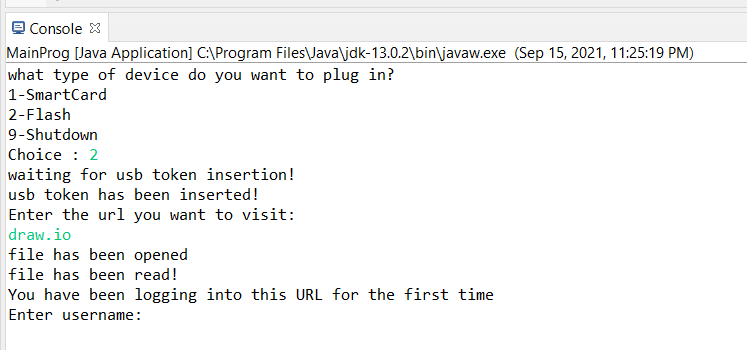
Picture 5



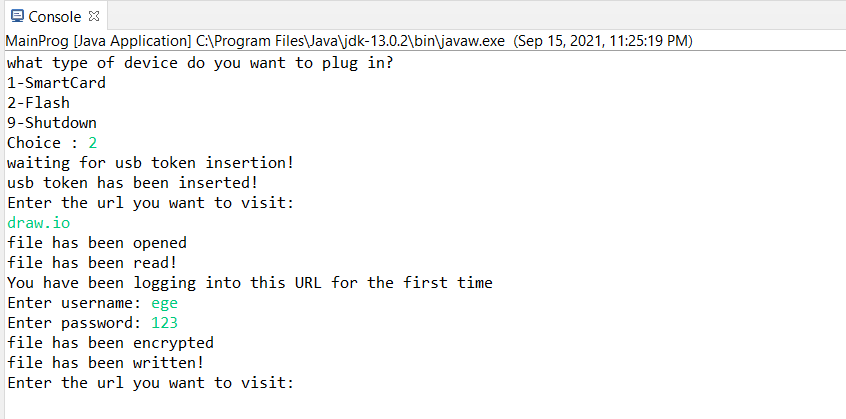
Picture 1(Storing user info USB device)



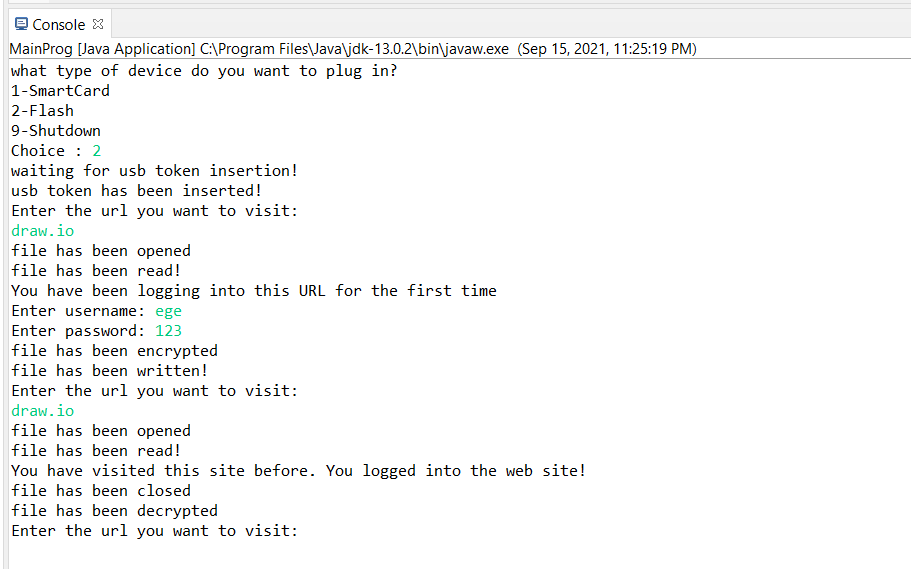
Picture 2(Storing user info USB device)



Picture 3(Storing user info USB device)



Picture 4(Storing user info USB device)



EGE ERGÜN