package week1;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

interface ReadWrite {

void write(String[] s);

void read(String[] s);

}

class Core implements ReadWrite {

//reader

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

public void write(String[] s) {

System.out.print("PASSWORD: ");

try {

s[0] = in.readLine();

} catch (IOException ex) {

ex.printStackTrace();

}

}

public void read(String[] s) {

//just outprints

System.out.println("Your password: " + s[0] + " has been accepted");

System.out.println("Welcome back! ");

}

}

package week1;

class Decorator implements ReadWrite {

private ReadWrite inner;

public Decorator(ReadWrite i) {

inner = i;

}

public void write(String[] s) {

inner.write(s);

}

public void read(String[] s) {

inner.read(s);

}

}

package week1;

class Scrambler extends Decorator {

public Scrambler(ReadWrite inner) {

super(inner);

}

public void write( String[] s ) {

super.write(s);

//this method is meant to change up everything

System.out.println("encrypting:");

StringBuilder sb = new StringBuilder(s[0]);

for (int i=0; i < sb.length(); i++) {

sb.setCharAt(i, (char)(sb.charAt(i) - 5));

}

s[0] = sb.toString();

}

public void read(String[] s) {

//this unchanges teh changes done

StringBuilder sb = new StringBuilder(s[0]);

for (int i=0; i < sb.length(); i++) {

sb.setCharAt(i, (char)(sb.charAt(i) + 5));

}

s[0] = sb.toString();

System.out.println( "decrypting:" );

super.read(s);

}

}

package week1;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class DecoratorSuper extends Decorator {

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

public DecoratorSuper(ReadWrite inner) {

super(inner);

}

public void write(String[] s) {

//this is jsut input

System.out.print("USERNAME: ");

try {

in.readLine();

} catch (IOException ex) {

ex.printStackTrace();

}

super.write( s );

}

public void read(String[] s) {

//this is also input

System.out.print("PASSWORD: ");

try {

in.readLine();

} catch (IOException ex) {

ex.printStackTrace();

}

super.read(s);

}

public static void main(String[] args) {

ReadWrite stream = new DecoratorSuper(new Scrambler(new Core()));

String[] str = {""};

stream.write(str);

System.out.println("main: " + str[0]);

stream.read(str);

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <<interface>> | <-- | Decorator | <-- | Scrambler |
| ReadWrite |  | |+ Decorator i: ReadWrite |  | |+ Scrambler(): inner: ReadWrite |
| s: String |  | |+ write s: String |  | |+ write() s: String |
| s: String |  | | + read s: String |  | |+ read() s: String |
| ^ |  | ^ |  |  |
| | |  | | |  |  |
| | |  | DecoratorSuper |  |  |
| Core |  | in: BuffedReader |  |  |
| in: BuffedReader |  | | + DecoratorSuper(): inner: ReadWrite |  |  |
| |+ write() s: String |  | |+ write() s: String |  |  |
| |+ read() s: String |  | |+ read() s: String |  |  |



