Slip 3

3.1

import java.util.Observable;

import java.util.Observer;

class WeatherData extends Observable {

private float temperature;

private float humidity;

private float pressure;

public void measurementsChanged() {

setChanged();

notifyObservers();

}

public void setMeasurements(float temperature, float humidity, float pressure) {

this.temperature = temperature;

this.humidity = humidity;

this.pressure = pressure;

measurementsChanged();

}

public float getTemperature() {

return temperature;

}

public float getHumidity() {

return humidity;

}

public float getPressure() {

return pressure;

}

}

class WeatherDisplay implements Observer {

private Observable observable;

public WeatherDisplay(Observable observable) {

this.observable = observable;

observable.addObserver(this);

}

@Override

public void update(Observable obs, Object arg) {

if (obs instanceof WeatherData) {

WeatherData weatherData = (WeatherData) obs;

display(weatherData.getTemperature(), weatherData.getHumidity(), weatherData.getPressure());

}

}

public void display(float temperature, float humidity, float pressure) {

System.out.println("Temperature: " + temperature + " °C");

System.out.println("Humidity: " + humidity + "%");

System.out.println("Pressure: " + pressure + " hPa");

System.out.println();

}

}

public class WeatherStation {

public static void main(String[] args) {

WeatherData weatherData = new WeatherData();

WeatherDisplay display1 = new WeatherDisplay(weatherData);

WeatherDisplay display2 = new WeatherDisplay(weatherData);

// Simulate weather changes

weatherData.setMeasurements(25.5f, 65.2f, 1012.3f);

weatherData.setMeasurements(28.0f, 70.5f, 1010.0f);

}

}

3.2

#Write a python program to make categorial values in numeric format

import pandas as pd

df=pd.read\_

csv('PlayTennis.csv')

print(df)

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

label=le.fit\_transform(df['Play Tennis'])

print(label)

df.drop("Play Tennis",axis=1, inplace=True)

df["Play Tennis"]=label

print(df

3.3

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 20px;

}

form {

max-width: 400px;

margin: 20px auto;

background: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

label {

display: block;

margin-bottom: 8px;

}

input {

width: 100%;

padding: 8px;

margin-bottom: 10px;

box-sizing: border-box;

}

button {

background-color: #4caf50;

color: #fff;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

</style>

</head>

<body>

<form id="loginForm">

<label for="email">Email:</label>

<input type="text" id="email" name="email" placeholder="Enter your email" required>

<button type="button" onclick="validateEmail()">Login</button>

</form>

<script>

function validateEmail() {

var email = document.getElementById('email').value;

var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailRegex.test(email)) {

alert('Please enter a valid email address');

return;

}

// If the email is valid, you can perform other actions, such as submitting the form.

alert('Email is valid!');

// Uncomment the next line to submit the form

// document.getElementById('loginForm').submit();

}

</script>

</body>

</html>