Slip 8

8.1

// GumballMachine class representing the Context

public class GumballMachine {

private State soldOutState;

private State noQuarterState;

private State hasQuarterState;

private State soldState;

private State currentState;

private int count = 0;

public GumballMachine(int numberOfGumballs) {

soldOutState = new SoldOutState(this);

noQuarterState = new NoQuarterState(this);

hasQuarterState = new HasQuarterState(this);

soldState = new SoldState(this);

this.count = numberOfGumballs;

if (numberOfGumballs > 0) {

currentState = noQuarterState;

} else {

currentState = soldOutState;

}

}

// Actions and behaviors

public void insertQuarter() {

currentState.insertQuarter();

}

public void ejectQuarter() {

currentState.ejectQuarter();

}

public void turnCrank() {

currentState.turnCrank();

currentState.dispense();

}

public void releaseBall() {

System.out.println("A gumball comes rolling out the slot...");

if (count != 0) {

count--;

}

}

// Other methods

public void refill(int numberOfGumballs) {

this.count = numberOfGumballs;

currentState = noQuarterState;

}

// Getters and setters

public State getSoldOutState() {

return soldOutState;

}

public State getNoQuarterState() {

return noQuarterState;

}

public State getHasQuarterState() {

return hasQuarterState;

}

public State getSoldState() {

return soldState;

}

public int getCount() {

return count;

}

public void setState(State state) {

this.currentState = state;

}

public State getState() {

return currentState;

}

public static void main(String[] args) {

GumballMachine gumballMachine = new GumballMachine(5);

// Test the Gumball Machine

gumballMachine.insertQuarter();

gumballMachine.turnCrank();

gumballMachine.insertQuarter();

gumballMachine.ejectQuarter();

gumballMachine.turnCrank();

gumballMachine.insertQuarter();

gumballMachine.turnCrank();

gumballMachine.insertQuarter();

gumballMachine.turnCrank();

gumballMachine.refill(10);

gumballMachine.insertQuarter();

gumballMachine.turnCrank();

}

}

8.2

#python program to implement Decision Tree whether or not to play Tennis

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

playTennis = pd.read\_csv('/home/pc10/ML\_Slips\_Solution/PlayTennis.csv')

from sklearn.preprocessing import LabelEncoder

Le=LabelEncoder()

playTennis['Outlook']=Le.fit\_transform(playTennis['Outlook'])

playTennis['Temperature']=Le.fit\_transform(playTennis['Temperature'] )

playTennis['Humidity']=Le.fit\_transform(playTennis['Humidity'])

playTennis['Wind']=Le.fit\_transform(playTennis['Wind'])

playTennis['Play Tennis']=Le.fit\_transform(playTennis['Play Tennis'])

print(playTennis)

y=playTennis['Play Tennis']

x=playTennis.drop(['Play Tennis'],axis=1)

from sklearn import tree

clf = tree.DecisionTreeClassifier(criterion='entropy')

clf=clf.fit(x,y)

fig = plt.figure(figsize=(25,20)) \_ = tree.plot\_tree(clf, filled=True)

plt.show()

8.3

const mysql = require('mysql');

// MySQL connection configuration

const connection = mysql.createConnection({

host: 'localhost',

user: 'your\_username',

password: 'your\_password',

database: 'your\_database\_name',

});

// Connect to MySQL server

connection.connect((err) => {

if (err) {

console.error('Error connecting to MySQL server:', err.message);

return;

}

console.log('Connected to MySQL server');

// Create a new database

const createDatabaseQuery = 'CREATE DATABASE IF NOT EXISTS mydatabase';

connection.query(createDatabaseQuery, (err) => {

if (err) {

console.error('Error creating database:', err.message);

return;

}

console.log('Database created or already exists');

// Use the newly created database

connection.changeUser({ database: 'mydatabase' }, (err) => {

if (err) {

console.error('Error selecting database:', err.message);

return;

}

console.log('Using database: mydatabase');

// Create a new table

const createTableQuery = `

CREATE TABLE IF NOT EXISTS users (

id INT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL

)

`;

connection.query(createTableQuery, (err) => {

if (err) {

console.error('Error creating table:', err.message);

return;

}

console.log('Table created or already exists');

// Close the MySQL connection

connection.end((err) => {

if (err) {

console.error('Error closing connection:', err.message);

return;

}

console.log('Connection closed');

});

});

});

});

});