**VEHICLE SIMULATOR**

**AIM:**

Suppose you are developing a simple game, that involves various types of vehicles. each vehicle has a unique name, maximum speed and a method to move(drive/ride,etc). To represent these different types of vehicles in your game, you decide to create hierarchy of classes using inheritance. you also decide to use an interface which defines method for moving each type of vehicle.

**PROGRAM**:

/\*\*

\*

\* @author 2162014

\*/

import java.util.Scanner;

// Define the interface for moving vehicles

interface Moveable {

void move();

}

// Define the base Vehicle class

class Vehicle {

String name;

int maxSpeed, fuelLevel, xPosition, yPosition;

public Vehicle(String name, int maxSpeed) {

this.name = name;

this.maxSpeed = maxSpeed;

this.fuelLevel = 100; // initialize fuel level to 100

// initialize position to (0, 0)

this.xPosition = 0;

this.yPosition = 0;

}

public void display() {

System.out.println("\n\*Vehicle Information\*");

System.out.println("\tName: " + name);

System.out.println("\tMax Speed: " + maxSpeed);

System.out.println("\tFuel Level: " + fuelLevel);

System.out.println("\tPosition: (" + xPosition + ", " + yPosition + ")");

}

}

// Define a Car class that extends Vehicle and implements Moveable

class Car extends Vehicle implements Moveable {

public Car(String name, int maxSpeed) {

super(name, maxSpeed);

}

public void move() {

Scanner cread = new Scanner(System.in);

System.out.println("\n\*Car Simulation\*\n");

System.out

.println("\*Instructions\* \n\tUse keys 'w', 'a', 's', 'd' to move the car " + name + " or 'x' to exit");

while (fuelLevel > 0) {

System.out.print("\nEnter key: ");

char move = cread.next().charAt(0);

switch (move) {

case 'w':

System.out.println("\tAction: Moved " + name + " forward");

fuelLevel -= 5;

yPosition += 1;

break;

case 'a':

System.out.println("\tAction: Turned " + name + " left");

fuelLevel -= 2;

xPosition -= 1;

break;

case 's':

System.out.println("\tAction: Moved " + name + " backward");

fuelLevel -= 5;

yPosition -= 1;

break;

case 'd':

System.out.println("\tAction: Turned " + name + " right");

fuelLevel -= 2;

xPosition += 1;

break;

case 'x':

System.out.println("\tAction: Exited car simulation");

return;

default:

System.out.println("\nInvalid Input Error: Valid keys are 'w', 'a', 's', 'd', and 'x'");

}

System.out.println("\t\*Car Status\*");

System.out.println("\t\tPosition: (" + xPosition + ", " + yPosition + ")");

if (fuelLevel < 0)

System.out.println("\t\tFuel Level: " + 0);

else

System.out.println("\t\tFuel Level: " + fuelLevel);

}

System.out.println("\nEmpty Fuel Status: " + name + " is out of fuel!");

}

}

// Define a Boat class that extends Vehicle and implements Moveable

class Boat extends Vehicle implements Moveable {

public Boat(String name, int maxSpeed) {

super(name, maxSpeed);

}

public void move() {

Scanner bread = new Scanner(System.in);

System.out.println("\n\*Boat Simulation\*\n");

System.out.println(

"Instructions: \n\tUse keys 'w', 'a', 's', 'd' to move the boat " + name

+ " or 'x' to exit simulation");

while (fuelLevel > 0) {

System.out.print("\nEnter key: ");

char move = bread.next().charAt(0);

switch (move) {

case 'w':

System.out.println("\tAction: Moved " + name + " forward");

fuelLevel -= 5;

yPosition += 1;

break;

case 'a':

System.out.println("\tAction: Turned " + name + " left");

fuelLevel -= 2;

xPosition -= 1;

break;

case 's':

System.out.println("\tAction: Moved " + name + " backward");

fuelLevel -= 5;

yPosition -= 1;

break;

case 'd':

System.out.println("\tAction: Turned " + name + " right");

fuelLevel -= 2;

xPosition += 1;

break;

case 'x':

System.out.println("\tAction: Exited boat simulation");

return;

default:

System.out.println("\nInvalid Input Error: Valid keys are 'w', 'a', 's', 'd', and 'x'");

}

System.out.println("\t\*Boat Status\*");

System.out.println("\t\tPosition: (" + xPosition + ", " + yPosition + ")");

if (fuelLevel < 0)

System.out.println("\t\tFuel Level: " + 0);

else

System.out.println("\t\tFuel Level: " + fuelLevel);

}

System.out.println("\nEmpty Fuel Status: " + name + " is out of fuel!");

}

}

// Define a Plane class that extends Vehicle and implements Moveable

class Plane extends Vehicle implements Moveable {

int altitude;

public Plane(String name, int maxSpeed) {

super(name, maxSpeed);

altitude = 0; // initialize altitude to 0

}

public void move() {

Scanner pread = new Scanner(System.in);

System.out.println("\n\*Plane Simulation\*\n");

System.out.println(

"Instructions: \n\tUse keys 'w', 'a', 's', 'd', 'q', 'z' to move the plane " + name

+ " or 'x' to exit simulation");

while (fuelLevel > 0) {

System.out.print("\nEnter key: ");

char move = pread.next().charAt(0);

switch (move) {

case 'w':

System.out.println("\tAction: Moved " + name + " forward");

fuelLevel -= 5;

yPosition += 1;

break;

case 'a':

System.out.println("\tAction: Turned " + name + " left");

fuelLevel -= 2;

xPosition -= 1;

break;

case 's':

System.out.println("\tAction: Moved " + name + " backward");

fuelLevel -= 5;

yPosition -= 1;

break;

case 'd':

System.out.println("\tAction: Turned " + name + " right");

fuelLevel -= 2;

xPosition += 1;

break;

case 'q':

System.out.println("\tAction: Climbed " + name);

altitude += 500;

fuelLevel -= 10;

break;

case 'z':

System.out.println("\tAction: Descended " + name);

altitude -= 500;

fuelLevel -= 10;

break;

case 'x':

System.out.println("\tAction: Exited plane simulation");

return;

default:

System.out.println("\nInvalid Input Error: Valid keys are 'w', 'a', 's', 'd', 'q', 'z', and 'x'.");

}

System.out.println("\t\*Plane Status\*");

System.out.println("\t\tAltitude: " + altitude);

System.out.println("\t\tPosition: (" + xPosition + ", " + yPosition + ")");

if (fuelLevel < 0)

System.out.println("\t\tFuel Level: " + 0);

else

System.out.println("\t\tFuel Level: " + fuelLevel);

}

System.out.println("\nFuel Status: " + name + " is out of fuel!");

}

}

// Define the main class that creates and moves the vehicles

public class VehicleGame {

public static void main(String[] args) {

int choice;

Scanner gread = new Scanner(System.in);

System.out.println("\nWelcome to the Vehicle Simulator!");

do {

System.out.println("\nChoose vehicle");

System.out.println("\t1. Car \n\t2. Boat \n\t3. Plane \n\t4. Quit");

System.out.print("Enter Choice: ");

choice = gread.nextInt();

switch (choice) {

case 1:

System.out.println("\nYou have chosen car!");

System.out.print("Enter the name of your car: ");

String carName = gread.next();

System.out.print("Enter the maximum speed of your car: ");

int carMaxSpeed = gread.nextInt();

Car car = new Car(carName, carMaxSpeed);

car.display();

car.move();

break;

case 2:

System.out.println("\nYou have chosen boat!");

System.out.print("Enter the name of your boat: ");

String boatName = gread.next();

System.out.print("Enter the maximum speed of your boat: ");

int boatMaxSpeed = gread.nextInt();

Boat boat = new Boat(boatName, boatMaxSpeed);

boat.display();

boat.move();

break;

case 3:

System.out.println("\nYou have chosen plane!");

System.out.print("Enter the name of your plane: ");

String planeName = gread.next();

System.out.print("Enter the maximum speed of your plane: ");

int planeMaxSpeed = gread.nextInt();

Plane plane = new Plane(planeName, planeMaxSpeed);

plane.display();

plane.move();

break;

case 4:

break;

default:

System.out.println("Invalid Choice Error: Valid choices are 1, 2, 3, and 4.");

break;

}

} while (choice < 4);

System.out.println("\nUntil next time!\n");

}

}

**RESULTS**:



    