**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**



**БИЕ ДААЛТЫН АЖЛЫН**

**ТАЙЛАН**

**Өгөгдлийн бүтэц ба алгоритм(F.CSm203)**

**2025-2026 оны хичээлийн жилийн намар**

**Бие даалтийн ажлын нэр:** Оюутны дүн бүртгэл

**Хичээл заасан багш:** Д.Батмөнх

**Бие даалтын ажил гүйцэтгэсэн:** Ж.Мэнд-Амгалан(B242270008)

**Лабораторын цаг:** 1-1

**Улаанбаатар хот**

Лабораторийн ажил – Car Parking System (Java, JUnit Test)

# 1. Ажлын зорилго

Энэхүү лабораторийн ажлын зорилго нь Java хэл дээр Stack өгөгдлийн бүтэц ашиглан автомашины зогсоолын системийг хэрэгжүүлэх болон JUnit framework ашиглан нэгж тест бичиж сурах явдал юм.

# 2. Ашигласан технологи ба орчин

- Java 17  
- VS Code IDE  
- JUnit 5 (org.junit.jupiter)  
- File I/O (BufferedReader, FileReader)  
- Stack өгөгдлийн бүтэц

# 3. Програмын бүтэц

Програм нь гурван үндсэн ангитай:  
1. Car – автомашины дугаарыг хадгалдаг энгийн класс.  
2. CarParking – автомашины ирэлт, гаралтыг удирдах гол класс.  
3. Main – файл уншиж, процесс гүйцэтгэх үндсэн класс.  
Мөн JUnit ашиглан бичсэн CarParkingTest нэртэй тестийн класс орно.

# 4. Алгоритмын тайлбар

Stack өгөгдлийн бүтэц ашигласнаар автомашин гарахад түр гаргасан машинуудыг түр зуурын stack-д хадгалж, дараа нь буцаан оруулдаг. Энэ нь бодит зогсоолын дарааллыг дуурайлган хэрэгжүүлсэн логик юм.

Үндсэн функцууд:  
- arrival(): машин орж ирэхэд Stack-д нэмнэ.  
- departure(): машин гарахад Stack-ээс хайж гаргана.  
- input(): файл уншиж, командын жагсаалт үүсгэнэ.  
- process(): командуудыг боловсруулна.  
- output(): одоо байгаа машинуудыг хэвлэнэ.

# 5. JUnit тестийн бүтэц ба зорилго

CarParkingTest класс нь дараах төрлийн нэгж тестүүдийг агуулна:  
- Arrival болон Departure функцуудын зөв ажиллагааг шалгах.  
- Stack-ийн дараалал зөв хадгалагдаж буй эсэх.  
- Файл унших функцийн ажиллагаа.  
- Edge case буюу онцгой нөхцөлүүд (хоосон гарааш, дүүрсэн гарааш, байхгүй машин гэх мэт).

# 6. Гүйцэтгэлийн үр дүн (жишээ гаралт)

Arrival UB11-11 -> There is room.  
Arrival BB22-22 -> There is room.  
Departure UB11-11 -> 0 cars moved out.  
Cars currently in garage:  
 - BB22-22

# 7. Дүгнэлт

Энэхүү лабораторийн ажлаар Stack өгөгдлийн бүтэц ашиглан автомашины зогсоолын системийг амжилттай хийж, Java I/O болон JUnit тестийн үндсэн ойлголтыг бататгав.

# 8. Эх код

--- Car.java ---

public class Car {  
 public String plate;  
  
 public Car(String plate) {  
 this.plate = plate;  
 }  
  
 @Override  
 public String toString() {  
 return this.plate;  
 }  
}

--- CarParking.java ---

import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.\*;  
  
public class CarParking {  
 public static final int MAX\_SIZE = 10;  
 public static Stack<Car> parking = new Stack<>();  
 public Stack<Car> temp = new Stack<>();  
  
 public List<String[]> input(String fileName) {  
 List<String[]> commands = new ArrayList<>();  
 try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {  
 String line;  
 while ((line = br.readLine()) != null) {  
 String[] parts = line.split(" ");  
 if (parts.length == 2)  
 commands.add(parts);  
 }  
 } catch (IOException e) {  
 System.out.println("Error reading file: " + e.getMessage());  
 }  
 return commands;  
 }  
  
 public void process(List<String[]> commands) {  
 for (String[] command : commands) {  
 String action = command[0];  
 String carNum = command[1];  
 Car car = new Car(carNum);  
  
 if (action.equalsIgnoreCase("A")) {  
 arrival(car);  
 } else if (action.equalsIgnoreCase("D")) {  
 departure(car);  
 }  
 }  
 }  
  
 public void arrival(Car car) {  
 System.out.print("Arrival " + car.plate + " -> ");  
 if (parking.size() < MAX\_SIZE) {  
 parking.push(car);  
 System.out.println("There is room.");  
 } else {  
 System.out.println("Garage full, this car cannot enter.");  
 }  
 }  
  
 public void departure(Car car) {  
 System.out.print("Departure " + car.plate + " -> ");  
 if (parking.isEmpty()) {  
 System.out.println("Garage is empty.");  
 return;  
 }  
  
 int moved = 0;  
 boolean found = false;  
  
 while (!parking.isEmpty()) {  
 Car top = parking.pop();  
 if (top.plate.equals(car.plate)) {  
 found = true;  
 break;  
 } else {  
 temp.push(top);  
 moved++;  
 }  
 }  
  
 if (found) {  
 System.out.println(moved + " cars moved out.");  
 } else {  
 System.out.println("This car not in the garage.");  
 }  
  
 while (!temp.isEmpty()) {  
 parking.push(temp.pop());  
 }  
 }  
  
 public void output() {  
 System.out.println("\nCars currently in garage:");  
 if (parking.isEmpty()) {  
 System.out.println("None.");  
 } else {  
 for (Car c : parking) {  
 System.out.println(" - " + c.plate);  
 }  
 }  
 }  
 public void clear() {  
 parking.clear();  
 temp.clear();  
 }  
}

--- Main.java ---

import java.util.List;  
  
public class Main {  
 public static void main(String[] args) {  
 CarParking carParking = new CarParking();  
 List<String[]> commands = carParking.input("src/cars.txt");  
 carParking.process(commands);  
 carParking.output();  
 }  
}

--- CarParkingTest.java ---

(JUnit тестийн бүрэн код хавсаргав.)