

Varianta C

Se consideră un fișier de intrare ce conține cod în limbajul de programare Java. Implementați un analizor lexical folosind FLEX care să realizeze următoarele funcționalități:

- 1) Identificarea și numărarea claselor definite.
- 2) Determinarea numărului de instanțe create pentru fiecare clasă.
- 3) Verificarea dacă sunt implementate metodele "get" și "set" pentru fiecare atribut al unei clase.

Fișierele de input Java vor respecta următoarele reguli și vor avea următoarea structură:

În interiorul fișierului, tot codul trebuie să se afle în interiorul uneia sau mai multor clase.

Clasa:

- 1) Declararea variabilelor se va face doar cu specificatorii de acces public sau private.
- 2) Variabilele sunt declarate doar la începutul clasei/metodei.
- 3) Clasa va avea următoarea ordine și reguli:
 - 3.1) Declarare attribute private/public, pot exista și clase fără attribute declarate
 - 3.2) Constructor, poate să fie implementat sau nu
 - 3.3) Metode de get/set
 - 3.4) Alte metode
 - 3.5) Metoda de main a cărei antet este mereu "public static void main(String[] args)":
 - Tipurile de date ce pot apărea sunt tipurile de date primitive și obiecte create din clasele declarate în fișier
 - În fișierele de input nu vor exista clase imbricate
 - Definirea claselor se va face fie ca: class NumeClasa sau public class NumeClasa
 - Într-un fișier nu pot exista mai multe clase definite cu "public class NumeClasa". Dacă există 2 sau mai multe astfel de clase, se va afișa doar un mesaj de eroare
 - Se vor contoriza doar instanțele create în interiorul metodelor main
 - În cazul array-urilor de obiecte se va specifica printr-o valoare de tipul int numărul de elemente din array

Exemple:

1) Input:

```
class Person {
    private String name;
    private int age;

    // constructor
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void displayInfo() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}

public class Student {
    private String nume;
    private int varsta;

    public Student() {
        this.nume = "None";
        this.varsta = 0;
    }

    public static void main(String[] args) {
        Student[] studenti = new Student[10];

        for (int i = 0; i < studenti.length; i++) {
            studenti[i] = new Student();
        }

        for (Student student : studenti) {
            System.out.println("Nume: " + student.nume + ", Varsta: " +
student.varsta);
        }
    }
}
```

Output:

Clase 2: Person, Student

Obiecte Person: 0

Obiecte Student: 10

Person: Metoda "get" nu este implementata pentru toate atributele

Metoda "set" nu este implementata pentru toate atributele

Student: Metoda "get" nu este implementata pentru toate atributele

Metoda "set" nu este implementata pentru toate atributele

2) Input:

```
public class MemoryAllocation {
    public static void main(String[] args) {
        int[] intArray = new int[10];
        int[] intArray2 = new int[10];
        String[] stringArray = new String[5];
        Employee e1 = new Employee(1);
        Employee e2 = new Employee(2);

    }
}

class Employee {
    // class member variable
    private int empId;
    private String empName;
    private String empDesignation;
    private String empCompany;

    public Employee(int eId) {
        this.empId = eId;
    }

    public int getEmpId() {
        return empId;
    }

    public void setEmpId(int empId) {
        this.empId = empId;
    }

    public String getEmpName() {
        return empName;
    }

    public void setEmpName(final String eName) {
        if (eName == null || eName.length() <= 0) {
            throw new IllegalArgumentException();
        }
    }

    public String getEmpDesignation() {
```

```
        return empDesignation;
    }

    public void setEmpDesignation(final String eDesignation) {
        this.empDesignation = eDesignation;
    }

    public String getEmpCompany() {
        return empCompany;
    }

    public void setEmpCompany(final String eCompany) {
        this.empCompany = eCompany;
    }
}
```

Output:

Clase 2: MemoryAllocation, Employee

Obiecte MemoryAllocation: 0

Obiecte Employee: 2

MemoryAllocation: Nu exista atribute

Employee: Metodele "get" si "set" sunt implementate

3) Input:

```
class Bike {  
  
    private String serialNumber;  
    private String model;  
  
    private double price;  
  
    public Bike(String serialNumber, String model, double price) {  
        this.serialNumber = serialNumber;  
        this.model = model;  
        this.price = price;  
    }  
  
    public String getSerialNumber() {  
        return serialNumber;  
    }  
}
```

```

    public void setSerialNumber(String serialNumber) {
        this.serialNumber = serialNumber;
    }

    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }
}

class BikeFactory {

    private int totalBikesProduced;

    private Bike[] bikesArr;

    public BikeFactory(int capacity){
        this.bikesArr = new Bike[capacity];
        this.totalBikesProduced = 0;
    }

    public int getTotalBikesProduced() {
        return totalBikesProduced;
    }

    public void setTotalBikesProduced(int totalBikesProduced) {
        this.totalBikesProduced = totalBikesProduced;
    }

    public void produceBike(String model, double price){
        if(getTotalBikesProduced() < (365*12/1.5)) {
            setTotalBikesProduced(getTotalBikesProduced() + 1);
            int b = getTotalBikesProduced() - 1;
            this.bikesArr[b] = new Bike("bike_" + b, model, price);
        }
        else System.out.println("Sorry, too many bikes!");
    }
}

```

```

public class BikeTest {

```

```
public static void main(String[] args) {  
  
    int bikesNo;  
  
    BikeFactory factory = new BikeFactory(10);  
  
}
```

Output:

Clase 3: Bike, BikeFactory, BikeTest

Obiecte Bike: 0

Obiecte BikeFactory: 1

Obiecte BikeTest: 0

Bike: Metodele "get" si "set" sunt implementate

BikeFactory: Metoda "get" nu este implementata pentru toate attributele

Metoda "set" nu este implementata pentru toate attributele