Lab Task 2: Declaration of methods and constructors.

OBJECTIVES:

i. To be familiar with parameterized constructor
ii. To be familiar with the static keyword
iii. To be familiar with the difference between class properties/methods and object properties and methods

Problem:

write a java program to create a Movie class with properties such as title, director, year (object variables) and totalMovies (class variable). Implement a parameterized constructor to initialize 2 objects. Include an object method to display the movie details and a class method to get the total number of movies.

Prerequisites:

i. Basic knowledge of Java programming language.
ii. Familiarity with object-oriented programming concepts.
iii. Understanding of variables, methods, and constructors in Java.

Learning Outcomes:

i. Understanding the concept of class variables and their usage.
ii. Differentiating between class variables and object variables.
iii. Understanding the role of constructors in initializing object properties.
iv. Implementing object methods to perform specific actions.
v. Utilizing class methods to access shared data or perform operations that don't depend on object state.

Problem Analysis:

The program aims to create a Movie class and demonstrate the usage of class variables, object variables, constructors, class methods, and object methods. It sets the properties of movies using a parameterized constructor and displays the movie details using an object method. Additionally, it calculates and displays the total number of movies using a class method.

Background Theory:

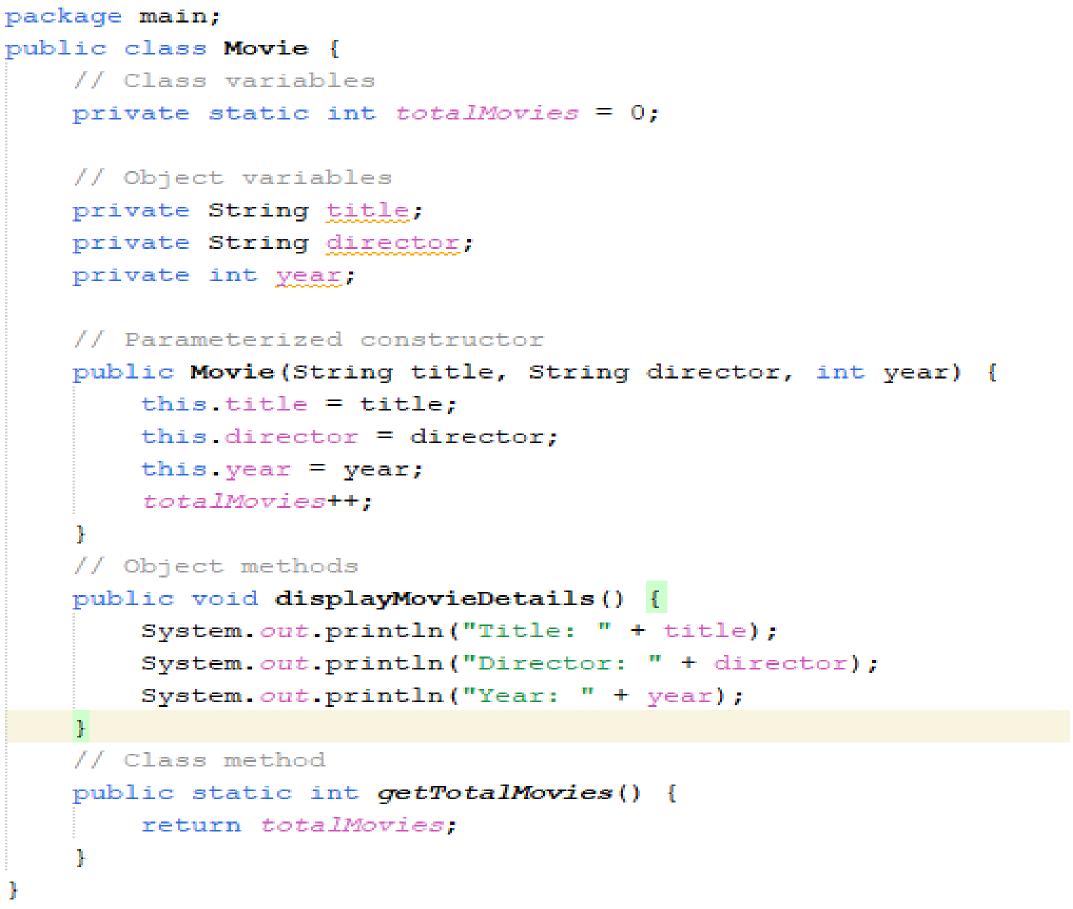
- Class variables are shared among all instances of a class. They are declared with the ‘static’ keyword.
- Object variables have unique values for each instance.
- Constructors are special methods used to initialize object properties when an object is created. It has the same name as the class name and it does not return anything. It can have parameters.

- Object methods perform actions specific to each object, whereas class methods are associated with the class and can be called without creating an object.
- Class methods are accessed using the class name. They are declared with the ‘static’ keyword.
- Object methods are accessed using the object instance.
- The main method serves as the entry point of the program.

Algorithm Design:

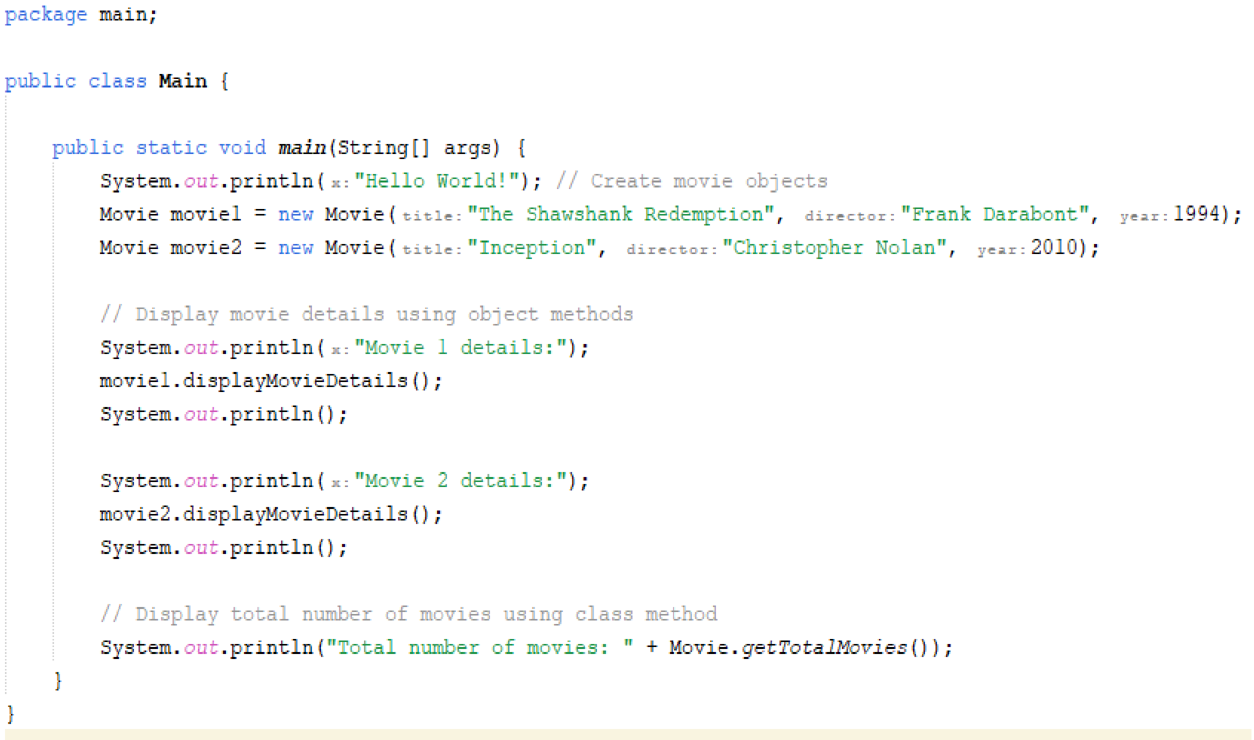
1. Create a Movie class.
2. Declare class variables, such as totalMovies, to keep track of the number of movies.
3. Declare object variables, such as title, director, and year, to store the movie properties.
4. Implement a parameterized constructor to set the movie properties and increment totalMovies.
5. Define an object method, such as displayMovieDetails(), to display the movie properties.
6. Implement a class method, such as getTotalMovies(), to return the total number of movies.
7. In the main function, create movie objects using the constructor and set their properties.
8. Use object methods to display the movie details.
9. Use the class method to display the total number of movies.

Code:



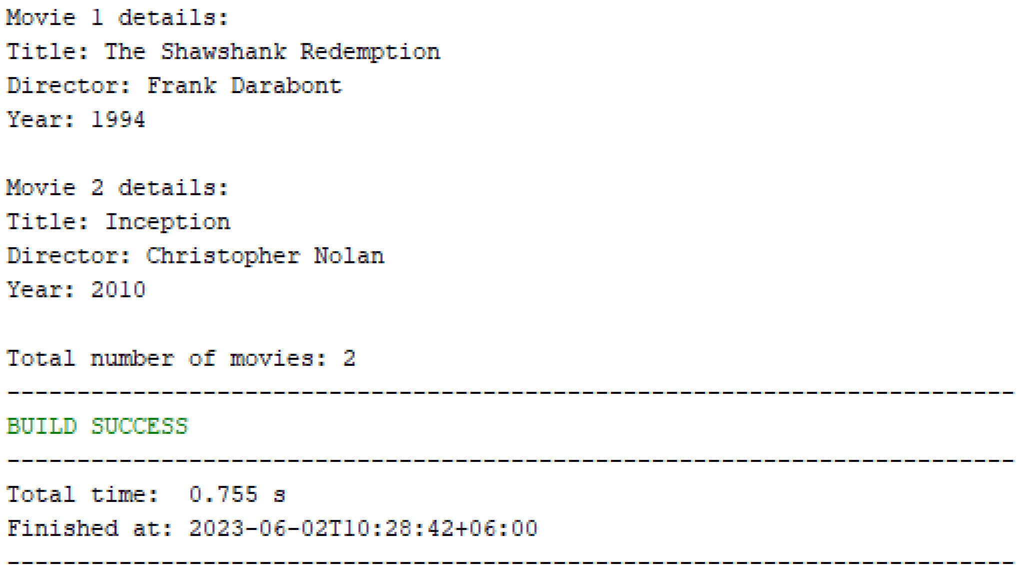
```java
package main;
public class Movie {
    // Class variables
    private static int totalMovies = 0;

    // Object variables
    private String title;
    private String director;
    private int year;

    // Parameterized constructor
    public Movie(String title, String director, int year) {
        this.title = title;
        this.director = director;
        this.year = year;
        totalMovies++;
    }
    // Object methods
    public void displayMovieDetails() {
        System.out.println("Title: " + title);
        System.out.println("Director: " + director);
        System.out.println("Year: " + year);
    }
    // Class method
    public static int getTotalMovies() {
        return totalMovies;
    }
}
```

Figure 1: Movie Class



```
package main;

public class Main {

    public static void main(String[] args) {
        System.out.println( x: "Hello World!"); // Create movie objects
        Movie movie1 = new Movie( title: "The Shawshank Redemption",  director: "Frank Darabont",  year: 1994);
        Movie movie2 = new Movie( title: "Inception",  director: "Christopher Nolan",  year: 2010);

        // Display movie details using object methods
        System.out.println( x: "Movie 1 details:");
        movie1.displayMovieDetails();
        System.out.println();

        System.out.println( x: "Movie 2 details:");
        movie2.displayMovieDetails();
        System.out.println();

        // Display total number of movies using class method
        System.out.println("Total number of movies: " + Movie.getTotalMovies());
    }
}
```

Figure 2: Main Class



```
Movie 1 details:
Title: The Shawshank Redemption
Director: Frank Darabont
Year: 1994

Movie 2 details:
Title: Inception
Director: Christopher Nolan
Year: 2010

Total number of movies: 2
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.755 s
Finished at: 2023-06-02T10:28:42+06:00
------------------------------------------------------------------------
```

Figure 3: Output

Practice Problems

1. Create an Employee class with properties such as name, age, designation, salary (object variables) and company name, company address (class variables). Implement a parameterized constructor to initialize 3 objects. Include an object method to display the employee details and a class method to display the total number of employees.

2. Create a Book class with properties such as title, author, year (object variables) and genre (class variable). Implement a parameterized constructor to initialize 3 objects. Include an object method to display the book details and a class method to display the total number of books.

3. Create a Student class with properties such as id, name, department, cgpa (object variables) and university (class variable). Implement a parameterized constructor to initialize 3 objects. Include an object method to display the student details and a class method to display the total number of students.