Practical 3

Aim: Design Creation Structure of a Vehicle using Prototype Design.

Description:
This program showcases object cloning by defining a Vehicle class with attributes like make, model, and year, implementing the Cloneable interface for cloning support. It prompts the user to input vehicle details, creates a prototype vehicle object with these details, and then clones it using the clone() method. The program then drives the cloned vehicle to demonstrate successful cloning. This approach allows for the creation of a new vehicle instance without the need to re-specify its details, highlighting the flexibility and utility of object cloning in Java.

Code:

1. Main.java

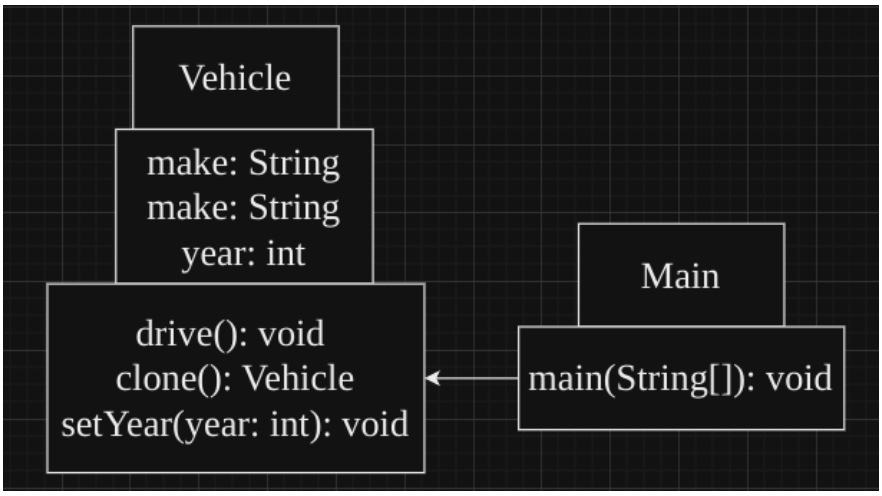
```
import java.util.Scanner;

// Define the Vehicle class
class Vehicle implements Cloneable {
    private String make;
    private String model;
    private int year;

    // Constructor
    public Vehicle(String make, String model, int year) {
        this.make = make;
        this.model = model;
        this.year = year;
    }

    // Method to drive the vehicle
    public void drive() {
        System.out.println("Driving the " + year + " " + make + " " + model);
    }

    // Method to clone the vehicle
    @Override
    public Vehicle clone() {
        try {
            return (Vehicle) super.clone();
        } catch (CloneNotSupportedException e) {
            throw new AssertionError(); // Should never happen
        }
    }

    // Method to set the year
    public void setYear(int year) {
        this.year = year;
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        // Create a scanner object for user input
        Scanner scanner = new Scanner(System.in);

        // Prompt user for vehicle details
        System.out.println("Enter the make of the vehicle:");
        String make = scanner.nextLine();

        System.out.println("Enter the model of the vehicle:");
        String model = scanner.nextLine();

        System.out.println("Enter the year of the vehicle:");
        int year = scanner.nextInt();

        // Create the prototype vehicle
        Vehicle prototypeVehicle = new Vehicle(make, model, year);

        // Clone the prototype vehicle
        Vehicle clonedVehicle = prototypeVehicle.clone();
        // No need to customize the year

        // Drive the cloned vehicle
        clonedVehicle.drive();
    }
}
```

Output:

```
go-d-code@code-valley:~/Roger/College/Design_lab_Codes/Prototype_design/1$ java Main
Enter the make of the vehicle:
Mahindra
Enter the model of the vehicle:
Scorpio
Enter the year of the vehicle:
2023
Driving the 2023 Mahindra Scorpio
```

Class Diagram:



Aim: Design Creation Structure of Electronic items using Prototype Design.

Description:
This program demonstrates the Prototype design pattern for creating electronic devices. It includes an abstract base class Electronics with concrete subclasses Television and Smartphone. The program prompts the user to input details such as the type (television or smartphone), brand, model, and year of the electronics. Depending on the type selected, it prompts for additional details like screen size for a television or processor for a smartphone. It then creates a prototype electronics object based on the user input, clones it, and allows the user to customize the cloned object's year. Finally, it displays the description of the cloned electronics. This implementation showcases how the Prototype pattern allows for the creation of new electronic devices with different configurations while maintaining a clear separation between object creation and representation.

Code:

1. Main.java

```
import java.util.Scanner;

// Abstract base class for electronics
abstract class Electronics implements Cloneable {
    private String brand;
    private String model;
    private int year;

    // Constructor
    public Electronics(String brand, String model, int year) {
        this.brand = brand;
        this.model = model;
        this.year = year;
    }

    // Getter method for brand
    public String getBrand() {
        return brand;
    }

    // Getter method for model
    public String getModel() {
        return model;
    }

    // Getter method for year
    public int getYear() {
        return year;
    }

    // Method to get description of the electronics
    public abstract String getDescription();

    // Clone method
    @Override
```

```
    public Electronics clone() {
        try {
            return (Electronics) super.clone();
        } catch (CloneNotSupportedException e) {
            throw new AssertionError(); // Should never happen
        }
    }

    // Method to set the year
    public void setYear(int year) {
        this.year = year;
    }
}

// Concrete subclass for television
class Television extends Electronics {
    private int screenSize;

    // Constructor
    public Television(String brand, String model, int year, int screenSize) {
        super(brand, model, year);
        this.screenSize = screenSize;
    }

    // Override method to get description
    @Override
    public String getDescription() {
        return "Television - " + getBrand() + " " + getModel() + ", Year: " + getYear() + ", Screen Size:
" + screenSize
                + " inches";
    }
}

// Concrete subclass for smartphone
class Smartphone extends Electronics {
    private String processor;

    // Constructor
    public Smartphone(String brand, String model, int year, String processor) {
        super(brand, model, year);
        this.processor = processor;
    }

    // Override method to get description
    @Override
    public String getDescription() {
        return "Smartphone - " + getBrand() + " " + getModel() + ", Year: " + getYear() + ", Processor:
" + processor;
    }
}

public class Main {
```

```
    public static void main(String[] args) {
        // Create a scanner object for user input
        Scanner scanner = new Scanner(System.in);

        // Prompt user for electronics details
        System.out.println("Enter the type of electronics (television/smartphone):");
        String type = scanner.nextLine();

        System.out.println("Enter the brand of the electronics:");
        String brand = scanner.nextLine();

        System.out.println("Enter the model of the electronics:");
        String model = scanner.nextLine();

        System.out.println("Enter the year of the electronics:");
        int year = scanner.nextInt();
        scanner.nextLine(); // Consume newline character

        // Create the prototype electronics object based on user input
        Electronics prototypeElectronics;
        if (type.equalsIgnoreCase("television")) {
            System.out.println("Enter the screen size of the television (in inches):");
            int screenSize = scanner.nextInt();
            prototypeElectronics = new Television(brand, model, year, screenSize);
        } else if (type.equalsIgnoreCase("smartphone")) {
            System.out.println("Enter the processor of the smartphone:");
            String processor = scanner.nextLine();
            prototypeElectronics = new Smartphone(brand, model, year, processor);
        } else {
            System.out.println("Invalid electronics type.");
            return;
        }

        // Clone the prototype electronics
        Electronics clonedElectronics = prototypeElectronics.clone();
        // Customize if necessary
        System.out.println("Enter the year of the cloned electronics:");
        int clonedYear = scanner.nextInt();
        clonedElectronics.setYear(clonedYear);

        // Display the cloned electronics
        System.out.println("Cloned electronics:");
        System.out.println(clonedElectronics.getDescription());
    }
}
```

Output:

```
go-d-code@code-valley:~/Roger/College/Design_lab_Codes/Prototype_design/2$ javac Main.java && java Main
Enter the type of electronics (television/smartphone):
smartphone
Enter the brand of the electronics:
Samsung
Enter the model of the electronics:
S23 Ultra
Enter the year of the electronics:
2023
Enter the processor of the smartphone:
Snapdragon
Enter the year of the cloned electronics:
2024
Cloned electronics:
Smartphone - Samsung S23 Ultra, Year: 2024, Processor: Snapdragon
```

Class Diagram: