NAME:
NIA:
GROUP:

Part B: Exercises

Time: 45 minutes

Instructions:

- No books or other resources allowed.
- Do not forget to write your name, NIA and group in every answer sheet of paper.

Exercise 1 (3.5 points out of 5)

We are requested to implement an application for managing auto parts for a technical service company.

- Any auto part has a manufacturer, a model, a description, a number (of pieces of such model in the stock) and a price per unit (in US dollars). For example, the following information would correspond to a car stereo:

```
Manufacturer: Pioneer
Model: DEHX6500BT
Description: Pioneer In-Dash AM/FM CD Player Car Stereo
Number: 5
Price: $129.97
```

- There are some subtypes of parts with specific features. For example, tires have a maximum recommended speed; fuel tanks need to specify if they are designed for petrol or diesel fuel; etc.

Program:

1. The `AutoPart` class, which represents a generic auto part.
   a. Include the appropriate attributes, according to the previous description.
   b. Provide a unique constructor that receives all the information related to the piece.
   c. Provide the `public void printDetails()` method, which prints on standard output the detailed description (all available information) of the piece, with the following format:
   ```
   Manufacturer: Pioneer
   Model: DEHX6500BT
   Description: Pioneer In-Dash AM/FM CD Player Car Stereo
   Number: 10
   Price: $129.97
   ```
   d. Provide the `getTypeOfPart()` method that is expected to return a String with the type of part. The type of part is not stored as an attribute but depends on the subtype of piece the object belongs to. This means that this method cannot be implemented in this generic class, but it must ensure that any `AutoPart` object has it.
2. The `RadioPart` class, which represents a particular type of auto part: a car radio. Car radios have as specific information the power output (in Watts) and if they allow Bluetooth connection or not.
   a. Declare the appropriate attributes, according to the description.
   b. Provide a unique constructor that receives all the information related to the radio part piece.
   c. Override the `public void print()` method, so that it prints on the standard output the complete information about the radio part, including both generic and specific description:
   ```
   Manufacturer: Pioneer
   Model: DEHX6500BT
   Description: Pioneer In-Dash AM/FM CD Player Car Stereo
   Number: 10
   Price: $129.97
   Bluetooth: yes
   Power output: 50W
   ```
   d. The `getTypeOfPart()` method must return “Radio” for any object of this class.

Exercise 2 (1.5 point out of 5)

Question A. Program the `GuiApp` class so that it presents the graphical user interface shown in the following figure:

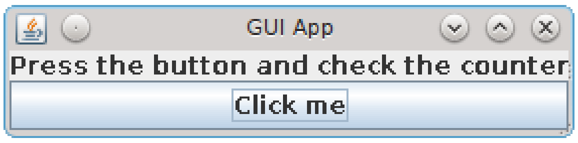

Figure 1 GUI App initial interface

Question B. Given the following skeleton for the `GuiAppListener` class, complete the gaps so that whenever the user clicks on the button that the listener is associated to, its text changes to show the counter value (as in the figures below).

```
public class GuiAppListener implements ActionListener {

   private int counter = 0;

   public void __________________________ ( _________________________________ ) {

      counter++;

      JButton pressedButton = ________________________________________________ ;

      _______________________________.setText("Pressed " + counter + " times!");

   }

}
```

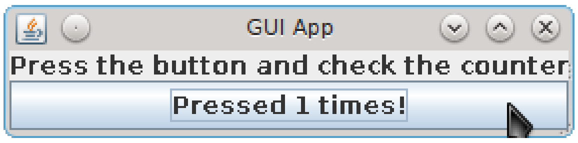

Figure 2 GUI App after pressing the button 1 time.

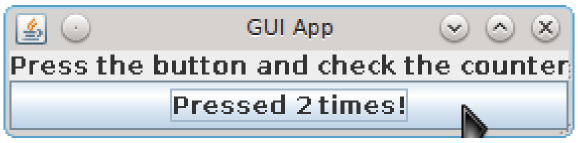

Figure 3 GUI App after pressing the button 2 times.

Solutions

Exercise 1

```
public abstract class AutoPart {

        private String manufacturer;
        private String model;
        private String description;
        private int number;
        private double price;

        public AutoPart(String manufacturer, String model, String description, int number, double price) {
                this.manufacturer = manufacturer;
                this.model = model;
                this.description = description;
                this.number = number;
                this.price = price;
        }

        public void print() {
                System.out.println("Manufacturer: " + manufacturer);
                System.out.println("Model: " + model);
                System.out.println("Description: " + description);
                System.out.println("Number: " + number);
                System.out.println("Price: $" + price);
        }

        public abstract String getTypeOfPart();

}
```

```
public class RadioPart extends AutoPart {

        private int power;
        private boolean bluetooth;

        public RadioPart(String manufacturer, String model, String description, int number, double price,
                                        int power, boolean bluetooth)
        {
                super(manufacturer, model, description, number, price);
                this.power = power;
                this.bluetooth = bluetooth;
        }

        public void print() {
                super.print();
                System.out.println("Bluetooth: " + bluetooth);
                System.out.println("Power output: " + power);
        }

        public String getTypeOfPart() {
                return "Radio";
        }

}
```

Exercise 2

```
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;


public class GuiApp {

	private void createGui() {
		JFrame frame = new JFrame("GUI App");
		JLabel label = new JLabel("Press the button and check the counter");
		JButton button = new JButton("Click me");

		GuiAppListener listener = new GuiAppListener();
		button.addActionListener(listener);

		frame.getContentPane().add(label, BorderLayout.NORTH);
		frame.getContentPane().add(button, BorderLayout.CENTER);

		frame.pack();
		frame.setVisible(true);
	}

	public static void main(String[] args) {
		GuiApp app = new GuiApp();
		app.createGui();
	}

}
```

```
class GuiAppListener implements ActionListener {

	private int counter = 0;

	public void actionPerformed(ActionEvent e) {
		counter++;
		JButton pressedButton = (JButton) e.getSource();
		pressedButton.setText("Pressed " + counter + " times!");
	}
}
```