Finding position in Java ME

This code snippet demonstrates how to retrieve device coordinates in Java ME. It uses `LocationProvider.getInstance()` to obtain `LocationProvider` instance, which represents a location-providing module, generating Locations, and `LocationProvider.getLocation()` method to retrieve current latitude and longitude.

Source file: FindingPosition.java

```
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.StringItem;
import javax.microedition.location.Coordinates;
import javax.microedition.location.Criteria;
import javax.microedition.location.Location;
import javax.microedition.location.LocationException;
import javax.microedition.location.LocationProvider;
import javax.microedition.midlet.MIDlet;

public class FindingPosition extends MIDlet implements CommandListener {

    private Display display;
    private Form form;
    private Command exitCommand;
    private Command refreshCommand;
    private StringItem text;

    private Location location;
    private LocationProvider locationProvider;
    private Coordinates coordinates;
    private Criteria criteria;

    /**
     * Constructor. Constructs the object and initializes displayables.
     */
    public FindingPosition() {
        form = new Form("Finding location.");

        exitCommand = new Command("Exit", Command.EXIT, 2);
        refreshCommand = new Command("Refresh", Command.OK, 1);

        text = new StringItem("Your position:", "\nPress \"Refresh\"");
        form.append(text);

        form.addCommand(exitCommand);
        form.addCommand(refreshCommand);
        form.setCommandListener(this);

        display = Display.getDisplay(this);
        display.setCurrent(form);

        criteria = new Criteria();
        criteria.setHorizontalAccuracy(500);
```


(C) Copyright Nokia 2014. All rights reserved.

```
        try {
            locationProvider = LocationProvider.getInstance(criteria);
        } catch (LocationException e) {
            //TODO: Handle location exception.
            return;
        }
    }

    /**
     * From MIDlet.
     * Called when the MIDlet is started.
     */
    public void startApp() {
        // No implementation required.
    }

    /**
     * From MIDlet.
     * Called to signal the MIDlet to enter the Paused state.
     */
    public void pauseApp() {
        // No implementation required.
    }

    /**
     * From MIDlet.
     * Called to signal the MIDlet to terminate.
     * @param unconditional whether the MIDlet has to be unconditionally
     * terminated
     */
    public void destroyApp(boolean unconditional) {
        // No implementation required
    }

    /**
     * From CommandListener.
     * Called by the system to indicate that a command has been invoked on a
     * particular displayable.
     * @param cmd the command that was invoked
     * @param displayable the displayable where the command was invoked
     */
    public void commandAction(Command c, Displayable d) {
        if (c == refreshCommand) {
            checkLocation();
        } else if (c == exitCommand) {
            notifyDestroyed();
        }
    }

    /**
     * Called to read current location.
     */
    private void checkLocation() {
        String string;
```


(C) Copyright Nokia 2014. All rights reserved.

```
        try {
            location = locationProvider.getLocation(60);
        } catch (Exception e) {
            //TODO: Handle exception.
            return;
        }

        coordinates = location.getQualifiedCoordinates();
        if (coordinates != null) {
            // Use coordinate information
            double lat = coordinates.getLatitude();
            double lon = coordinates.getLongitude();
            string = "\nLatitude : " + lat + "\nLongitude : " + lon;
        } else {
            string = "Location API failed";
        }
        text.setText(string);
    }
}
```

Postconditions

After user press "refresh", MIDlet will show current coordinates.

Supplementary material

You can see source file and executable application in attached zip archive. Archive is available for download at Finding_position_in_Java_ME.zip

(C) Copyright Nokia 2014. All rights reserved.