Lab 03 Exercise Solutions

Lab03 Exercises

Exercise 1

In PlacemarkActivity - we have the following hard coded strings:

PlacemarkActivity

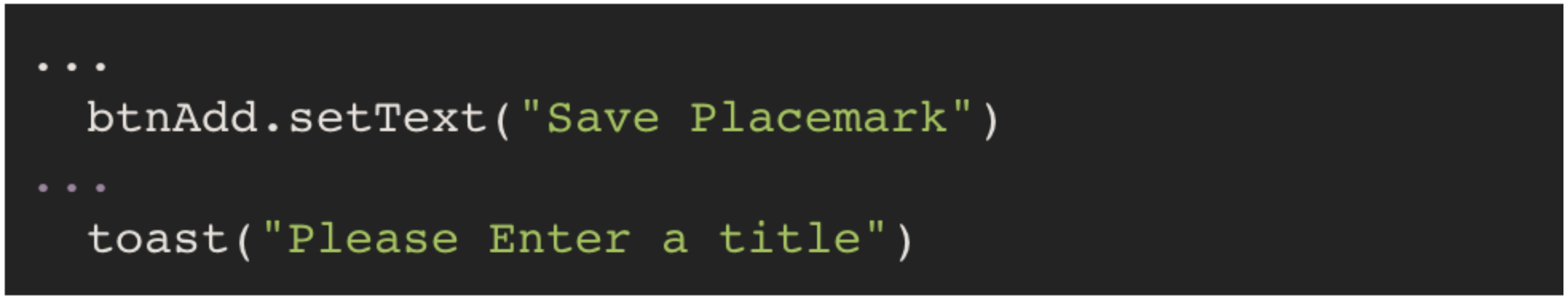
```
...
  btnAdd.setText("Save Placemark")
...
  toast("Please Enter a title")
```

These strings should be in the strings.xml resource file. Move them in there, and figure out how to load the strings in PlacemarkActivity.

For guidance, see:

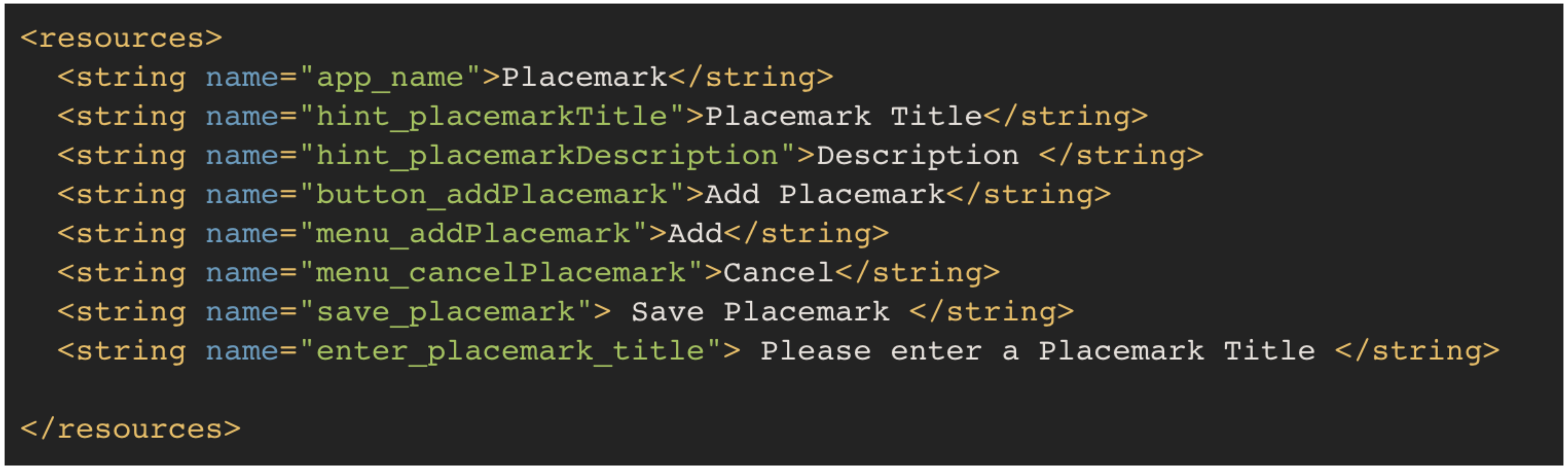
- https://developer.android.com/guide/topics/resources/string-resource.html
- https://stackoverflow.com/questions/44871481/how-can-i-access-values-from-strings-xml-in-kotlin-android

PlacemarkActivity

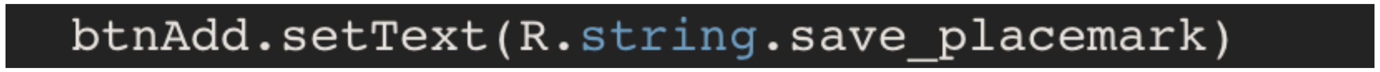


```
...
  btnAdd.setText("Save Placemark")
...
  toast("Please Enter a title")
```

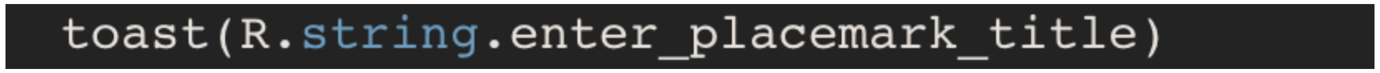
strings.xml



```
<resources>
  <string name="app_name">Placemark</string>
  <string name="hint_placemarkTitle">Placemark Title</string>
  <string name="hint_placemarkDescription">Description </string>
  <string name="button_addPlacemark">Add Placemark</string>
  <string name="menu_addPlacemark">Add</string>
  <string name="menu_cancelPlacemark">Cancel</string>
  <string name="save_placemark"> Save Placemark </string>
  <string name="enter_placemark_title"> Please enter a Placemark Title </string>

</resources>
```



```
  btnAdd.setText(R.string.save_placemark)
```



```
 toast(R.string.enter_placemark_title)
```

Exercise 2

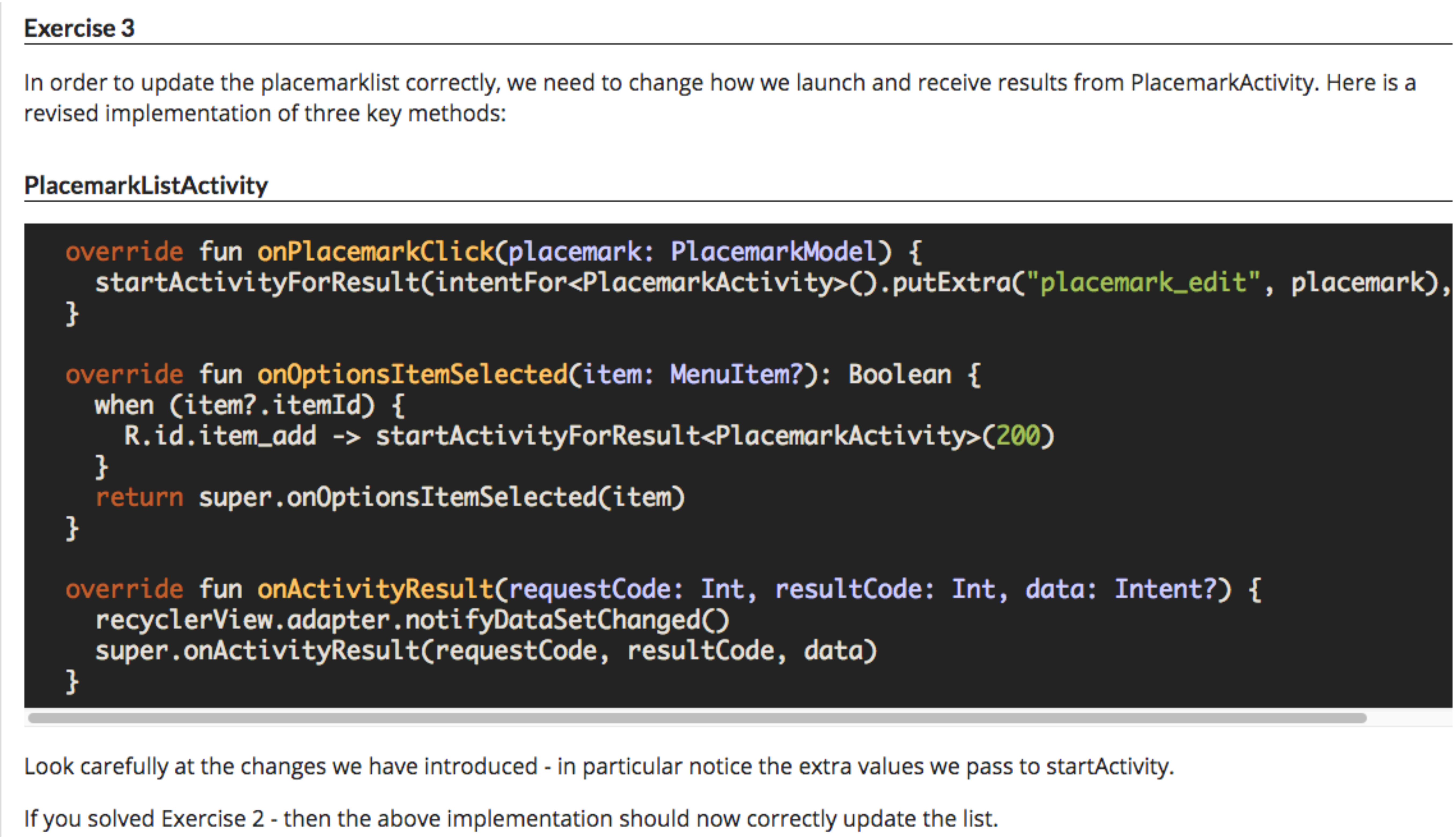
When you select a placemark from PlacemarkList Activity - it presents the activity in the PlacemarkActivity. If you edit the activity, and press 'Create' - a new activity is created and appended to the list.

Change this behaviour along the following lines:

- When PlacemarkActivity is opened in this way - the button label 'add placemark' should be named 'save placemark'. Make sure you are creating the string resources in strings.xml
- Pressing the button should update the existing placemark with the newly entered values, not create new one.

We have new features in PlacemarkStore to enable you to implement this (the update method)

If you complete the implementation, you will notice that even if you update a Placemark's contents, the new contents will not be displayed in PlacemarkList. However, if you consult the logs you may be able to see if the values are actually changed.



Exercise 3

In order to update the placemarklist correctly, we need to change how we launch and receive results from PlacemarkActivity. Here is a revised implementation of three key methods:

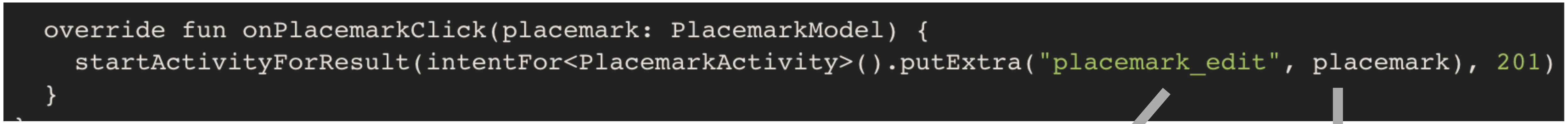
PlacemarkListActivity

```
  override fun onPlacemarkClick(placemark: PlacemarkModel) {
    startActivityForResult(intentFor<PlacemarkActivity>().putExtra("placemark_edit", placemark),
  }

  override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    when (item?.itemId) {
      R.id.item_add -> startActivityForResult<PlacemarkActivity>(200)
    }
    return super.onOptionsItemSelected(item)
  }

  override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    recyclerView.adapter.notifyDataSetChanged()
    super.onActivityResult(requestCode, resultCode, data)
  }
```

Look carefully at the changes we have introduced - in particular notice the extra values we pass to startActivity.

If you solved Exercise 2 - then the above implementation should now correctly update the list.

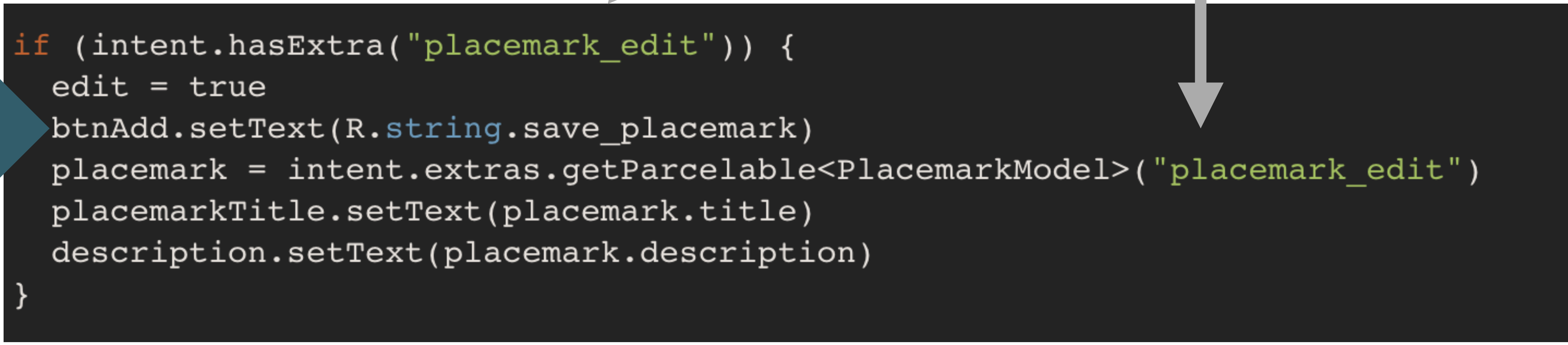
PlacemarkListActivity



```
  override fun onPlacemarkClick(placemark: PlacemarkModel) {
    startActivityForResult(intentFor<PlacemarkActivity>().putExtra("placemark_edit", placemark), 201)
  }
```

placemark sent to PlacemarkActivity

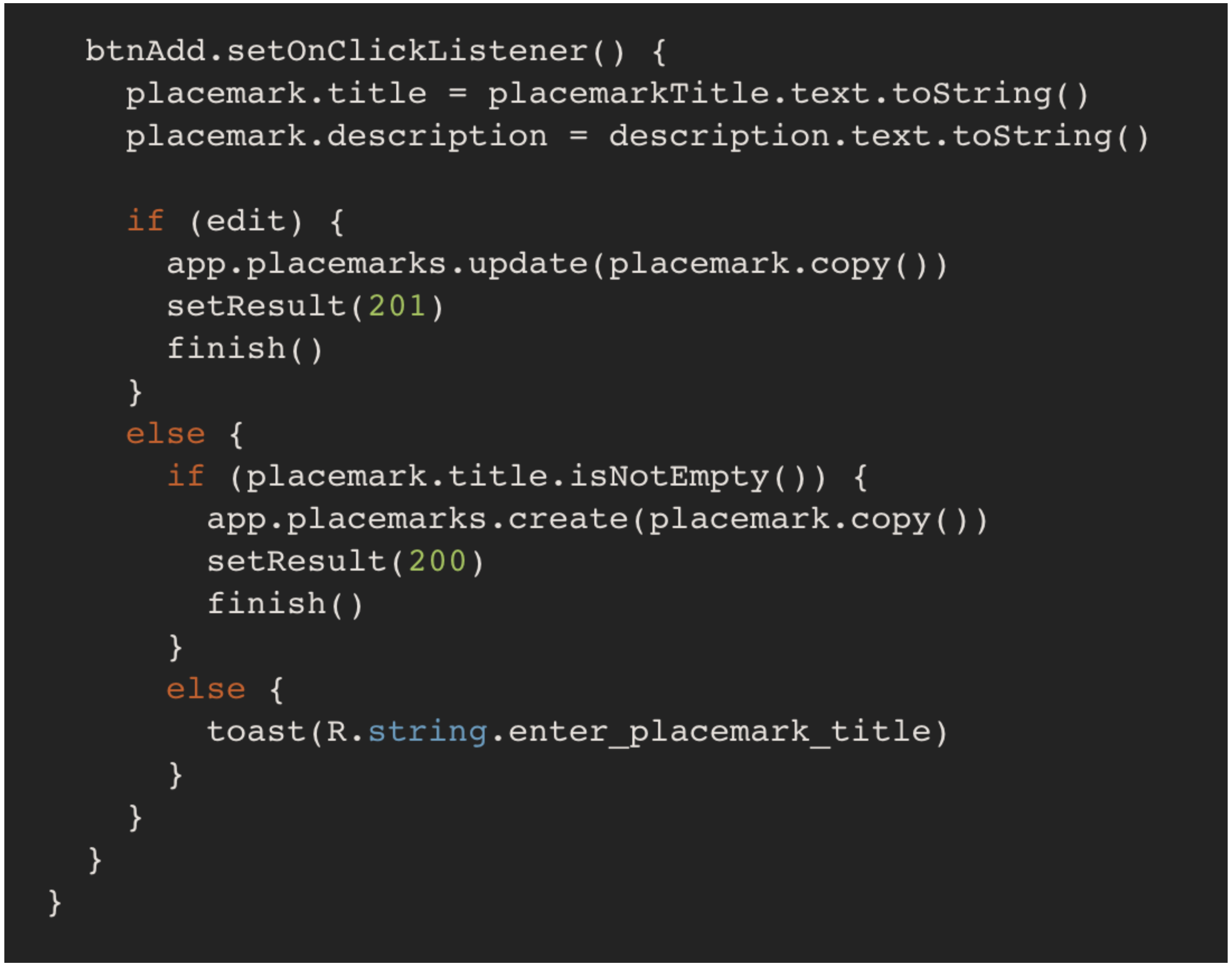
Set title to ‘Save” if pacemaker detected



```
if (intent.hasExtra("placemark_edit")) {
  edit = true
  btnAdd.setText(R.string.save_placemark)
  placemark = intent.extras.getParcelable<PlacemarkModel>("placemark_edit")
  placemarkTitle.setText(placemark.title)
  description.setText(placemark.description)
}
```

PlacemarkActivity

PlacemarkActivity addButon event handler

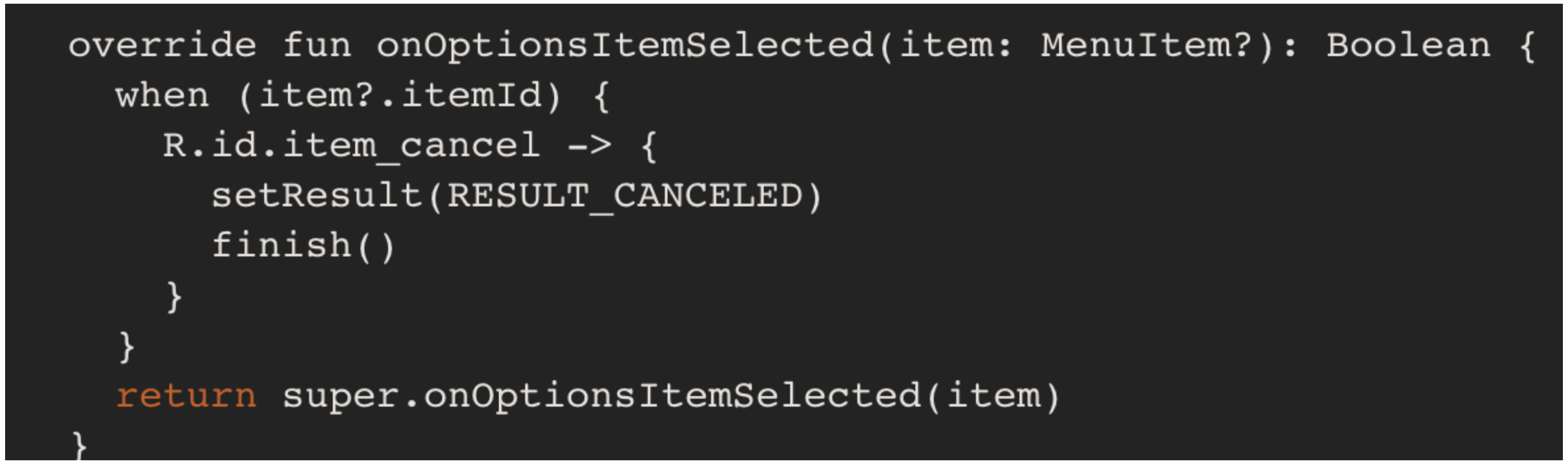


```
  btnAdd.setOnClickListener() {
    placemark.title = placemarkTitle.text.toString()
    placemark.description = description.text.toString()

    if (edit) {
      app.placemarks.update(placemark.copy())
      setResult(201)
      finish()
    }
    else {
      if (placemark.title.isNotEmpty()) {
        app.placemarks.create(placemark.copy())
        setResult(200)
        finish()
      }
      else {
        toast(R.string.enter_placemark_title)
      }
    }
  }
}
```

Check to see if we are **editing** an existing placemark

If we are - call **update()**

Otherwise call **create()**

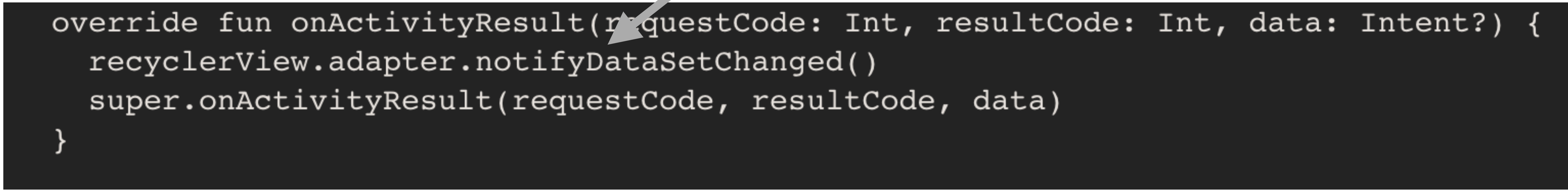


```
override fun onOptionsItemSelected(item: MenuItem?): Boolean {
  when (item?.itemId) {
    R.id.item_cancel -> {
      setResult(RESULT_CANCELED)
      finish()
    }
  }
  return super.onOptionsItemSelected(item)
}
```

If cancel pressed, end activity without any update

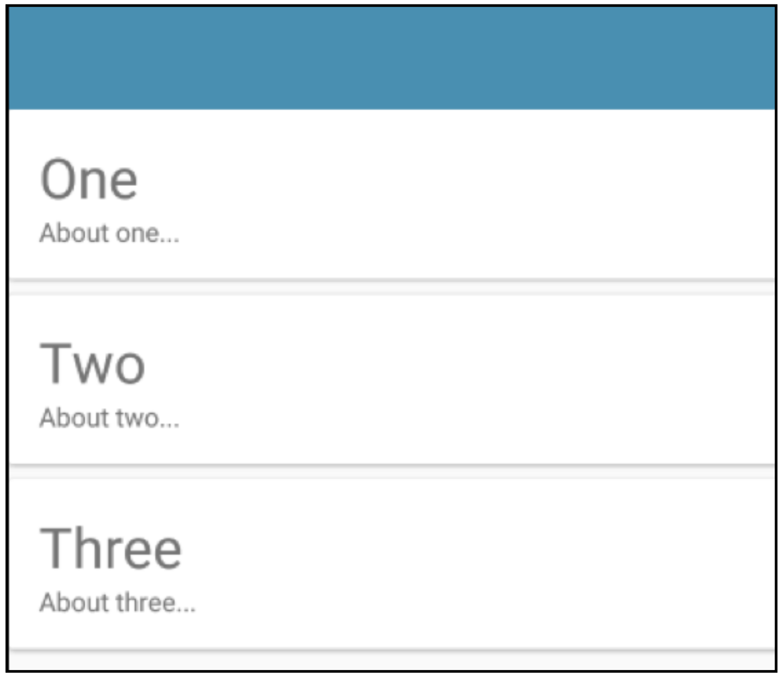
PlacemarkListActivity

When PlacemarkActivity finishes, and PlacemarkListActivity resumes, refresh the adapter



```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
  recyclerView.adapter.notifyDataSetChanged()
  super.onActivityResult(requestCode, resultCode, data)
}
```

This will redraw the lists, updating contents



One
About one...

Two
About two...

Three
About three...