



CS/SE Individual Practical

Stephen Gilmore
November 11, 2011

School of Informatics, University of Edinburgh

Common Java problems: raw types

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.food);
    String[] foodItems = {"Place1", "Place2"};
    setListAdapter(new ArrayAdapter<T>(this, android.R.layout.simple_list_item_1, foodItems));
    getListView().setTextFilterEnabled(true);
}
```

Problems 0 errors, 8 warnings, 0 others

Warnings (8 items)

- ArrayAdapter is a raw type. References to generic type ArrayAdapter<T> should be parameterized
- The import android.content.ActivityNotFoundException is never used
- The import android.content.Intent is never used
- The import android.net.Uri is never used
- The import android.util.Log is never used
- The value of the field DBHelper.TAG is not used
- The value of the field ForumActivity.buttonHotel is not used
- Type safety: The constructor ArrayAdapter(Context, int, Object[]) belongs to the raw type ArrayAd...

Quick fix: a bad suggestion

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.food);
    String[] foodItems = {"Place1", "Place2"};
    setListAdapter(new ArrayAdapter<T>(this, android.R.layout.simple_list_item_1, foodItems));
    getListView().setTextFilterEnabled(true);
}
```

Type safety: The constructor ArrayAdapter(Context, int, Object[]) belongs to the raw type ArrayAd...
References to generic type ArrayAdapter<T> should be parameterized

1 quick fix available:

- Add @SuppressWarnings('unchecked') to 'onCreate!'

Check the constructor documentation

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.food);
    String[] foodItems = {"Place1", "Place2"};
    setListAdapter(new ArrayAdapter<T>(this, android.R.layout.simple_list_item_1, foodItems));
    getListView().setTextFilterEnabled(true);
}
```

android.widget.ArrayAdapter.ArrayAdapter(Context context, int textViewResourceId, T[] objects)

public ArrayAdapter (Context context, int textViewResourceId, T[] objects)

Since: API Level 1

Constructor

Parameters

- context The current context.
- textViewResourceId The resource ID for a layout file containing a TextView to use when instantiating views.
- objects The objects to represent in the ListView.

Check the class documentation

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.food);
    String[] foodItems = {"Place1", "Place2"};
    setListAdapter(new ArrayAdapter<T>(this, android.R.layout.simple_list_item_1, foodItems));
    getListView().setTextFilterEnabled(true);
}
```

android.widget.ArrayAdapter<T>

A concrete BaseAdapter that is backed by an array of arbitrary objects. By default this class expects that the provided resource id references the constructors that also takes a field id. That field id should reference a TextView in the larger layout resource.

However the TextView is referenced, it will be filled with the toString() of each object in the array. You can add lists or arrays of custom object what text will be displayed for the item in the list.

To use something other than TextViews for the array display, for instance, ImageViews, or to have some of data besides toString() results fill the type of view you want.

Summary

Inherited Constants

From interface android.widget.Adapter

What type of array are we adapting?

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.food);
    String[] foodItems = {"Place1", "Place2"};
    setListAdapter(new ArrayAdapter<T>(this, android.R.layout.simple_list_item_1, foodItems));
    getListView().setTextFilterEnabled(true);
}
```

String[] foodItems = {"Place1", "Place2"};

In this case, T = String

Type parameter added

protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.food);
 String[] foodItems = {"Place1", "Place2"};
 setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, foodItems));
 getListView().setTextFilterEnabled(true);
}

Problems 0 errors, 6 warnings, 0 others
 Warnings (6 items)
 The import android.content.ActivityNotFoundException is never used
 The import android.content.Intent is never used
 The import android.net.Uri is never used
 The import android.util.Log is never used
 The value of the field DBHelper.TAG is not used
 The value of the field ForumActivity.buttonHotel is not used

Two fewer warnings

CS/SE Individual Practical 7 / 24

Raw type in method header

```
public void onNothingSelected(AdapterView parent) {  

    // Do nothing.  

}
```

Problems 0 errors, 5 warnings, 0 others
 Warnings (5 items)
 AdapterView is a raw type. References to generic type AdapterView<T> should be parameterized
 AdapterView is a raw type. References to generic type AdapterView<T> should be parameterized
 The import android.content.Intent is never used
 The import android.net.Uri is never used
 The import uni.ip.AccommodationTab.AccommodationOnItemSelectedListener is never used

CS/SE Individual Practical 8 / 24

Consult the Java documentation

```
public void onNothingSelected(AdapterView parent) {  

    // Do nothing.  

}
```

android.widget.AdapterView<T>
 An AdapterView is a view whose children are determined by an Adapter.
 See ListView, GridView, Spinner and Gallery for commonly used subclasses of AdapterView.

Summary

CS/SE Individual Practical 9 / 24

Adapter doesn't work here

```
public void onNothingSelected(AdapterView<Adapter> parent) {  

    // Do nothing.  

}
```

Problems 3 errors, 4 warnings, 0 others
 Errors (3 items)
 Adapter cannot be resolved to a type
 Name clash: The method onNothingSelected(AdapterView<Adapter>) of type AccommodationTab.AccommodationOnItemSelectedListener must implement the inherited abstract method AdapterView.
 The type AccommodationTab.AccommodationOnItemSelectedListener must implement the inherited abstract method AdapterView.
 Warnings (4 items)
 AdapterView is a raw type. References to generic type AdapterView<T> should be parameterized
 The import android.content.Intent is never used
 The import android.net.Uri is never used
 The import uni.ip.AccommodationTab.AccommodationOnItemSelectedListener is never used

CS/SE Individual Practical 10 / 24

Adapter is an interface

```
public void onNothingSelected(AdapterView parent) {  

    // Do nothing.  

}
```

android.widget.Adapter
 Known Indirect Subclasses
 ArrayAdapter<T>, BaseAdapter, CursorAdapter, HeaderViewListAdapter, ListAdapter, R
 SimpleCursorAdapter, SpinnerAdapter, WrapperListAdapter

Class Overview

CS/SE Individual Practical 11 / 24

T is not a class

```
public void onNothingSelected(AdapterView<T> parent) {  

    // Do nothing.  

}
```

Problems 3 errors, 4 warnings, 0 others
 Errors (3 items)
 Name clash: The method onNothingSelected(AdapterView<T>) of type AccommodationTab.AccommodationOnItemSelectedListener
 T cannot be resolved to a type
 The type AccommodationTab.AccommodationOnItemSelectedListener must implement the inherited abstract method AdapterView.
 Warnings (4 items)
 AdapterView is a raw type. References to generic type AdapterView<T> should be parameterized
 The import android.content.Intent is never used
 The import android.net.Uri is never used
 The import uni.ip.AccommodationTab.AccommodationOnItemSelectedListener is never used

CS/SE Individual Practical 12 / 24

Object: the goto class

```

public void onNothingSelected(AdapterView<Object> parent) {
    // Do nothing.
}
    
```

Problems 3 errors, 4 warnings, 0 others

Description

- Errors (3 items)
 - Bound mismatch: The type Object is not a valid substitute for the bounded parameter <T extends AdapterView> of the type AdapterView
 - Name clash: The method onNothingSelected(AdapterView<Object>) of type AccommodationTab.AccommodationOnItemSelectedList
 - The type AccommodationTab.AccommodationOnItemSelectedListener must implement the inherited abstract method AdapterView.C
- Warnings (4 items)
 - AdapterView is a raw type. References to generic type AdapterView<T> should be parameterized
 - The import android.content.Intent is never used
 - The import android.net.Uri is never used
 - The import uni.ip.AccommodationTab.AccommodationOnItemSelectedListener is never used

CS/SE Individual Practical 13 / 24

T extends Adapter doesn't work

```

public void onNothingSelected(AdapterView<T extends Adapter> parent) {
    // Do nothing.
}
    
```

Problems 7 errors, 4 warnings, 0 others

Description

- Errors (7 items)
 - Adapter cannot be resolved to a type
 - Incorrect number of arguments for type AdapterView<T>; it cannot be parameterized with arguments <T, Adapter>
 - Syntax error on token "T", delete this token
 - Syntax error on token "extends", expected
 - Syntax error, insert ";" to complete ClassBody
 - T cannot be resolved to a type
 - The type AccommodationTab.AccommodationOnItemSelectedListener must implement the inherited abstract method AdapterView.C
- Warnings (4 items)
 - AdapterView is a raw type. References to generic type AdapterView<T> should be parameterized
 - The import android.content.Intent is never used
 - The import android.net.Uri is never used
 - The import uni.ip.AccommodationTab.AccommodationOnItemSelectedListener is never used

CS/SE Individual Practical 14 / 24

"?" — the wild card parameter

```

public void onNothingSelected(AdapterView<?> parent) {
    // Do nothing.
}
    
```

Problems 0 errors, 3 warnings, 0 others

Description

- Warnings (3 items)
 - The import android.content.Intent is never used
 - The import android.net.Uri is never used
 - The import uni.ip.AccommodationTab.AccommodationOnItemSelectedListener is never used

CS/SE Individual Practical 15 / 24

"Quick fix" would have worked here

Problems 0 errors, 4 warnings, 0 others

Description

- Warnings (4 items)
 - AdapterView is a raw type. References to generic type AdapterView<T> should be parameterized
 - The import android.co
 - The import android.ne
 - The import uni.ip.Acc

Infer Generic Type Arguments

Infer type arguments for references to generic types and remove unnecessary casts.

- Assume clone() returns an instance of the receiver type
- Have unconstrained type arguments raw (rather than inferring <?>)

Uncheck this box

Preview > OK Cancel

CS/SE Individual Practical 16 / 24

Same result

```

public void onNothingSelected(AdapterView<?> parent) {
    // Do nothing.
}
    
```

Problems 0 errors, 3 warnings, 0 others

Description

- Warnings (3 items)
 - The import android.content.Intent is never used
 - The import android.net.Uri is never used
 - The import uni.ip.AccommodationTab.AccommodationOnItemSelectedListener is never used

CS/SE Individual Practical 17 / 24

Setting Java compiler preferences

Preferences

type filter text

- General
- Android
- Ant
- Help
- Install/Update
- Java
 - Appearance
 - Build Path
 - Code Style
 - Compiler
 - Building
 - Errors/Warnings
 - Task Tags
 - Debug
 - Editor
 - Installed JREs
 - JUnit
 - Properties Files Editor
 - Maven
 - Mylyn
 - Run/Debug
 - Team
 - Usage Data Collector
 - Validation
 - WindowBuilder
 - XML

Errors/Warnings

Select the severity level for the following optional Java compiler problems:

type filter text (use - to filter on preference values, e.g. --ignore or --off)

- Code style
 - Non-static access to static member: Warning
 - Indirect access to static member: Ignore
 - Unqualified access to instance field: Ignore
 - Undocumented empty block: Ignore
 - Access to a non-accessible member of an enclosing type: Ignore
 - Method with a constructor name: Warning
 - Parameter assignment: Ignore
 - Non-externalized strings (missing/unused \$NON-NLS\$ tag): Ignore
 - Method can be static: Ignore
 - Method can potentially be static: Ignore
- Potential programming problems
 - Name shadowing and conflicts
 - Deprecated and restricted API
 - Unnecessary code
 - Generic types
 - Annotations

Include 'assert' in null analysis

Treat above errors like fatal compiler errors (make compiled code not executable)

CS/SE Individual Practical 18 / 24

Changing defaults

CS/SE Individual Practical 19 / 24

Potential programming problems

CS/SE Individual Practical 20 / 24

Tighter checking

CS/SE Individual Practical 21 / 24

Bug found

CS/SE Individual Practical 22 / 24

Don't forget to add logging ...

public final class Log

Summary: Constants | Methods | Inherited Methods | [Expand All] Since: API Level 1

Log

extends Object

java.lang.Object ↳ android.util.Log

Class Overview

API for sending log output.

Generally, use the Log.v() Log.d() Log.i() Log.w() and Log.e() methods.

The order in terms of verbosity, from least to most is ERROR, WARN, INFO, DEBUG, VERBOSE. Verbose should never be compiled into an application except during development. Debug logs are compiled in but stripped at runtime. Error, warning and info logs are always kept.

Tip: A good convention is to declare a TAG constant in your class:

```
private static final String TAG = "MyActivity";
```

and use that in subsequent calls to the log methods.

Tip: Don't forget that when you make a call like

```
Log.v(TAG, "index=" + i);
```

CS/SE Individual Practical 23 / 24

Deeper analysis (resource leaks ...)

Klocwork

Productivity tools for software developers
Static Analysis - Peer Code Review - Refactoring

Bulletproof your Android Code
Build Secure, Reliable Apps & Devices
Built-in Support for Android APIs
Reduce your Costs

Solutions for Android Application and Device Development

With support for C, C++, and Java, Klocwork provides a robust solution for all software teams doing application, kernel, or other lower-level software development.

Android Application Development

Klocwork has built a robust Java static analysis engine that provides deep analysis of Java source code. Whether you're a Fortune 100 device manufacturer or an individual application developer, your Android apps need to be rock solid. An app that crashes or contains security vulnerabilities won't last long in the Android application universe.

Deep static analysis
Automatically find these types of defects in your code:
• Resource Leaks

Android Integration
Coupled with built-in knowledge of the Android APIs, developers will obtain accurate analysis of their

Customizable
Developers can use a rich set of tools that allows them to extend Klocwork's analysis and write their own

Risk Free 30-Day Trial
Fields marked (*) are required

First Name *

Last Name *

Company *

Email Address *

Phone Number *

CS/SE Individual Practical 24 / 24