

CE0825a - Object Oriented Programming II 5: Internet Access, SWT Menus

James A Sutherland

Abertay University

Monday, 8th February 2016

Getting a URL

Nice and easy: URL object

```
import java.net.URL;  
...  
URL u=new URL("http://www.sutherland.pw/");  
// can throw MalformedURLException
```

Retrieving text

```
URL url = new URL("http://www.sutherland.pw/");
URLConnection con = url.openConnection();
InputStream is = con.getInputStream();
InputStreamReader isr = new
    InputStreamReader(is);
BufferedReader input=new BufferedReader(isr);
String line;
while((line=input.readLine())!=null) {
    System.out.println(line);
}
```

Crash Reporting

Real-world: catch an `Exception e`, send `e.printStackTrace()` to a server via HTTP(S).

Free version of Windows Error Reporting, years earlier!

Throw away any exception that happens during that, there's nothing useful to do with it anyway! (Actually, dump it to the console, just in case we're debugging it at the time.)

A Real Example

```
catch (Exception e) {  
  try {  
    new URL("https://xxxxx/dumprep.php?" +  
      "product=wlce&v="+URLEncoder.encode(  
        BuildTime.stamp, "UTF-8")+  
      "&rep="+URLEncoder.encode(e.toString()+  
        "\n"+Utils.join("\n", e.getStackTrace()),  
        "UTF-8")).openStream();  
  } catch (Exception ne) {  
    ne.printStackTrace();  
  }  
}
```

General SWT Pattern

Many SWT components take some sort of option or style value as an `int` argument. These are predefined constants within the SWT class; sometimes, these can be combined with a mathematical OR operation, the Java `|` operator.

For example:

```
new Text(shell, SWT.BORDER | SWT.WRAP | SWT.MULTI);
```

Can make a big difference, for example `Button` can be `RADIO` style.

If you aren't using any specific style, put `SWT.NONE`.

SWT Menus

In week 3, we got an SWT Display and put a Shell in it. Now let's look at menus:

- Menu bar: `new Menu(shell,SWT.BAR)`
- Menu items: `new MenuItem(menubar,SWT.CASCADE)`
- Menus: `new Menu(shell,SWT.DROP_DOWN)`
- Entries: `new MenuItem(menu,SWT.PUSH)`

SWT Menus

In week 3, we got an SWT Display and put a Shell in it. Now let's look at menus:

- Menu bar: `new Menu(shell,SWT.BAR)`
- Menu items: `new MenuItem(menubar,SWT.CASCADE)`
- Menus: `new Menu(shell,SWT.DROP_DOWN)`
- Entries: `new MenuItem(menu,SWT.PUSH)`

SWT Menus

In week 3, we got an SWT Display and put a Shell in it. Now let's look at menus:

- Menu bar: `new Menu(shell, SWT.BAR)`
- Menu items: `new MenuItem(menubar, SWT.CASCADE)`
- Menus: `new Menu(shell, SWT.DROP_DOWN)`
- Entries: `new MenuItem(menu, SWT.PUSH)`

SWT Menus

In week 3, we got an SWT Display and put a Shell in it. Now let's look at menus:

- Menu bar: `new Menu(shell, SWT.BAR)`
- Menu items: `new MenuItem(menubar, SWT.CASCADE)`
- Menus: `new Menu(shell, SWT.DROP_DOWN)`
- Entries: `new MenuItem(menu, SWT.PUSH)`

Linking MenuItems

A MenuItem is created as part of a Menu already, and gets its text from the `.setText()` method.

A CASCADE MenuItem will have a Menu associated with it via a `menuItem.setMenu(menu)` call.

Tip: Windows (sometimes) underlines shortcut keys, for example File. Do this in Java by prefixing that letter with `&`.

Reacting to MenuItems

You attach a `SelectionListener` to the `MenuItem` like this:

```
exit.addSelectionListener(new
    SelectionListener(){
        @Override
        public void
            widgetSelected(SelectionEvent e) {
                mainshell.dispose();
            }
        @Override
        public void
            widgetDefaultSelected(SelectionEvent
            e) {
                // Doesn't need to do anything
            }
    });
```

Anonymous Inner Classes

Wait ... `SelectionListener` is an interface, not a Class, so how can we instantiate it?

Add implementations of all the required methods inline after the `new`, wrapped in `{ }`.

Actually, we could also use `SelectionAdapter`, which gives us (empty) 'implementations' of both methods, so we only need to override the one we want. (That avoids the empty `widgetDefaultSelected` method from earlier.)

Combining Web and SWT

Make the Help menu item open up a web browser showing a URL:

```
Shell helper=new Shell();
helper.setText("Help is here");
helper.setLayout(new FillLayout());
Browser browser = new Browser(helper,SWT.NONE);
browser.setUrl("http://imgtfy.com/?q=help");
helper.open();
```

Lab Task 5

Download an image from the web, saving it to a local file.

Get some (working!) menus populated, including an item to open an HTML help page loaded from a local file rather than a web page. (Rather than use a file: URL, handle it in Java yourself!)

Finished early and want to show off a bit? Add a CoolBar too...