

Hướng dẫn lập trình cơ bản với Android

Part 11

B5: Chỉnh sửa file strings.xml trong res\value:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Example 3</string>
    <string name="work_hint">Enter the work
here</string>
    <string name="hour_edit">Hour</string>
    <string name="minute_edit">Minute</string>
    <string name="button_content">Add work</string>
</resources>
```

B6: Time to coding. Đi tới src\at.exam tạo một class mới là CustomViewGroup với nội dung sau:

Mã:

```
package at.exam;

import android.content.Context;
import android.view.LayoutInflater;
import android.widget.CheckBox;
import android.widget.LinearLayout;
import android.widget.TextView;

public class CustomViewGroup extends LinearLayout {
    public CheckBox cb;
    public TextView workContent;
    public TextView timeContent;
    public CustomViewGroup(Context context) {
        super(context);
    }
}
```

```

        //Sử dụng LayoutInflater để gán giao diện trong
list.xml cho class này
        LayoutInflater li = (LayoutInflater)
this.getContext()

        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        li.inflate(R.layout.list, this, true);

        //Lấy về các View qua Id
        cb = (CheckBox) findViewById(R.id.check_work);
        workContent = (TextView)
findViewById(R.id.work_content);
        timeContent = (TextView)
findViewById(R.id.time_content);
    }
}

```

Đoạn code trên giúp ta định nghĩa giao diện của custom ViewGroup mới dựa trên file list.xml. Mọi người cũng có thể tạo giao diện bằng code, ko cần sử dụng XML nhưng sẽ phức tạp hơn và mình cũng ko giới thiệu ở đây.

B7: Tạo 1 class Work cũng trong at.exam để thể hiện công việc:
Mã:

```

package at.exam;

public class Work {
    private String workContent;
    private String timeContent;
    private boolean isChecked;

    public Work(String workContent, String timeContent)
    {
        this.workContent = workContent;
        this.timeContent = timeContent;
        isChecked = false;
    }

    public String getContent() {
        return workContent;
    }
}

```

```
public String getTime() {  
    return timeContent;  
}  
  
public void setChecked(boolean isChecked) {  
    this.isChecked = isChecked;  
}  
  
public boolean isChecked() {  
    return isChecked;  
}  
}
```

Code rất đơn giản nên mình sẽ không chú thích nữa.

B8: Chúng ta đã tạo xong custem ViewGroup, bây giờ chính là lúc sử dụng. Tạo 1 class mới tên là ListWorkApdapter trong at.exam:

Mã:

```
package at.exam;  
  
import java.util.ArrayList;  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.CheckBox;  
import android.widget.CompoundButton;  
import android.widget.TextView;  
import  
android.widget.CompoundButton.OnCheckedChangeListener;  
  
public class ListWorkAdapter extends  
ArrayAdapter<Work>{  
    ArrayList<Work> array;  
    int resource;  
    Context context;
```

```

    public ListWorkAdapter(Context context, int
textViewResourceId,
        ArrayList<Work> objects) {
        super(context, textViewResourceId, objects);
        this.context = context;
        resource = textViewResourceId;
        array = objects;
    }

    //Phương thức xác định View mà Adapter hiển thị, ở
    đây chính là CustomViewGroup
    //Bắt buộc phải Override khi kế thừa từ
    ArrayAdapter
    @Override
    public View getView(int position, View convertView,
ViewGroup parent) {
        View workView = convertView;

        if (workView == null) {
            workView = new
CustomViewGroup(getContext());
        }

        //Lấy về đối tượng Work hiện tại
        final Work work = array.get(position);

        if (work != null) {
            TextView workContent = ((CustomViewGroup)
workView).workContent;
            TextView timeContent = ((CustomViewGroup)
workView).timeContent;
            CheckBox checkWork = ((CustomViewGroup)
workView).cb;

            //Set sự kiện khi đánh dấu vào checkbox
            trên list
            checkWork.setOnCheckedChangeListener(new
OnCheckedChangeListener() {

                @Override

```

```

        public void
onCheckedChanged(CompoundButton buttonView,
                    boolean isChecked) {
            work.setChecked(isChecked);
        }
    });

    //Lấy về nội dung cho TextView và CheckBox
    dựa vào đối tượng Work hiện tại
    workContent.setText(work.getContent());
    timeContent.setText(work.getTime());
    checkWork.setChecked(work.isChecked());
}
return workView;
}
}

```

ListWorkAdapter sẽ được sử dụng thay thế cho ArrayAdapter được bind với ListView. Thông thường ArrayAdapter chỉ cho hiển thị String bằng TextView, nhưng với việc kế thừa và override phương thức getView, ta có thể định nghĩa lại hiển thị cho các thành phần của ListView.