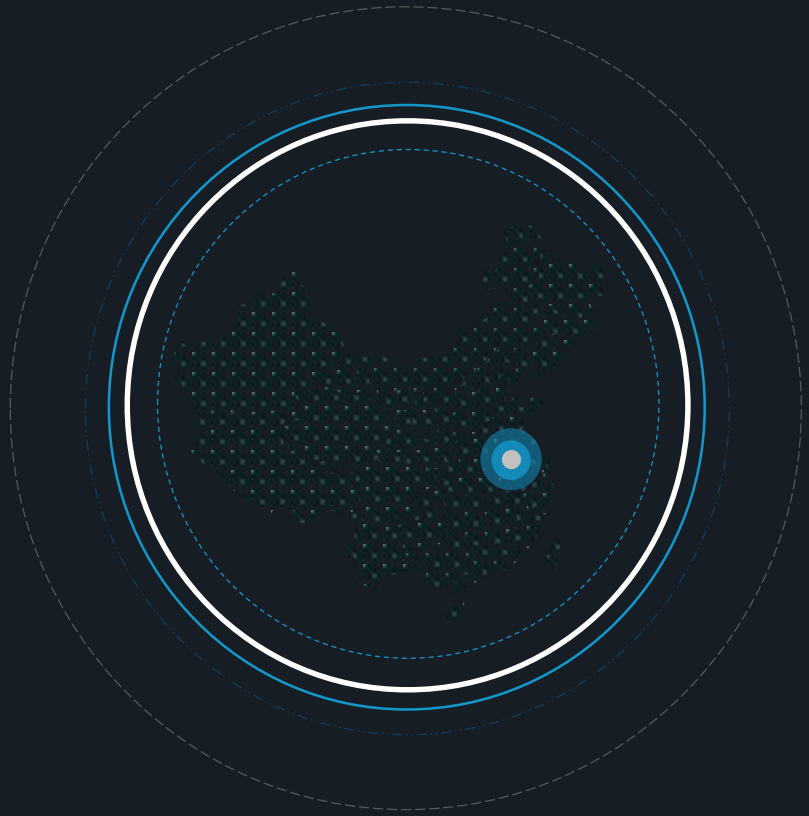
Physicas Lab Calculator



Group 1 : Smart Boys
王多 柴峥豪 贾萧松 李明泽
2017.6.1

Introduction

- Compulsory course -- Physics Lab

  -- Data operations are usually rather complicated

  -- The methods are similar

Introduction

- The solution -- Physics Lab Calculator

  -- Convenience

  Raw Data → Results

  -- Universality

  Customized Data Size

Task Allocation

- User Interface

  Adapter Design -- Wang Duo

  Data Acquisition and Transmission -- Jia Xiaosong

  Layout and Data Display -- Li Mingze

- Calculation

  Chai Zhenghao

CONTENTS

by Li Mingze

Overall Explanation

1

by Li Mingze

Layout and Data Display

2

Layout and Data Display

by Li Mingze

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:layout_centerHorizontal="true"
    android:id="@+id/Text1"
    android:textSize="30sp"
    android:text="请选择一个实验"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:padding="20dp"
    android:id="@+id/Button1"
    android:layout_below="@id/Text1"
    android:text="测量刚体转动惯量"
    android:textSize="20sp"/>
```

Layout and Data Display

by Li Mingze

Input of the number of measurement and the original data are the same!

```
    <TextView
        android:id="@+id/settinglist_text"
        android:layout_width="180sp"
        android:layout_height="30sp"
        android:textSize="25sp"/>
    <EditText
        android:id="@+id/settinglist_edittext"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:textColorHint="@color/colorPrimaryDark"
        android:inputType="number"
        android:imeOptions="actionNext"/>

</LinearLayout>
```

Layout and Data Display

by Li Mingze

```
HashMap<String,String>[] maps = new HashMap[num];

for (int i = 0; i < num; i++) {
    maps[i] = new HashMap<String,String>();
    maps[i].put("variable_name", names[i]);
    maps[i].put("variable_num",Double.toString( values[i]));
    list.add(maps[i]);
}
SimpleAdapter listAdapter = new SimpleAdapter(this,list,R.layout.showlist_sub_lmz,new String[]
    {"variable_name","variable_num"},new int[]{R.id.textView_li1,R.id.textView_li2});
//listAdapter = new Adapter(this,list);
```

```
    <TextView
        android:id="@+id/textView_li1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"/>

    <TextView
        android:id="@+id/textView_li2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"/>

</LinearLayout>
```

by Wang Duo

Adapter Design 3

by Wang Duo

Adaptor Design

Simple Adaptor

-- key-value pairs

variable_name → name[0]	variable_value → value[0]
variable_name → name[1]	variable_value → value[1]
variable_name → name[2]	variable_value → value[2]
variable_name → name[3]	variable_value → value[3]
... ...	

list

Adaptor Design

by Wang Duo

◈Simple Adaptor

-- Construction

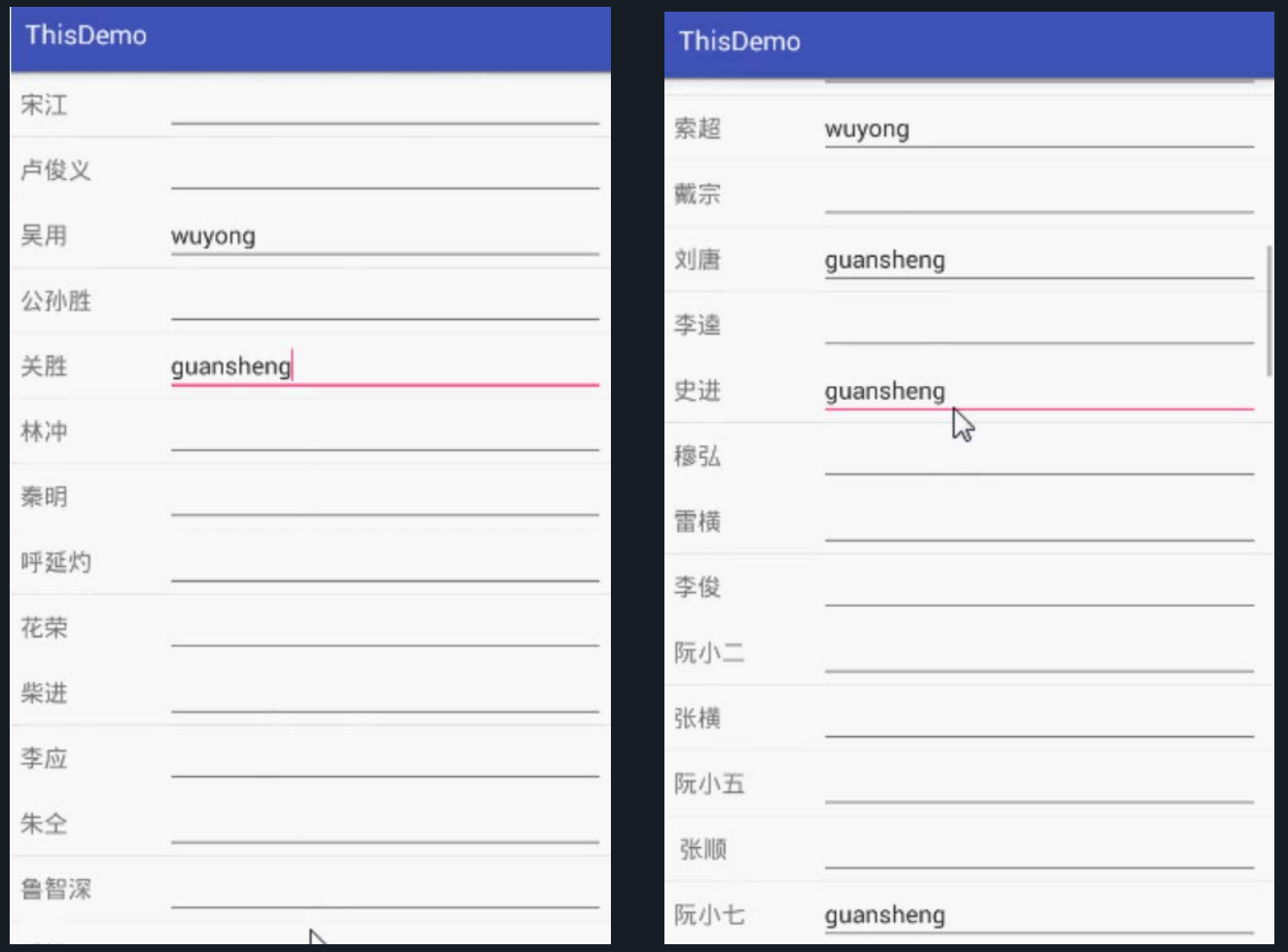
```
SimpleAdapter listAdapter = new SimpleAdapter(this, list, R.layout.setting_list_sub,
        new String[]{"variable_name","variable_value"},new int[]{R.id.name_text,R.id.value_text});
```

by Wang Duo

Adaptor Design

◈Problem of SimpleAdapter

-- Item-recycling mechanism



Adaptor Design

by Wang Duo

◈Custom Adaptor

-- Storing

-- Attaching

-- Reassigning

```
public  class Adapter extends BaseAdapter{
    ....
private String[] data;//Container of inputs
    ....
  @Override
public View getView(int position, View convertView, ViewGroup parent) {
    vh = new ViewHolder();//get the objects attached to the widgets
    vh.textview = (TextView) convertView.findViewById(R.id.settinglist_text);
    vh.edittext = (EditText) convertView.findViewById(R.id.settinglist_edittext);
    ....
    vh.edittext.setTag(position);//attach the widget with its position in the list
    ....
    vh.edittext.addTextChangedListener(new TextWatcher() {
        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            int position = (int)vh.edittext.getTag();
                data[position] =s.toString();//store the information according to position
        }
    });
    if(!TextUtils.isEmpty(data[position]+"")){
        vh.edittext.setText(data[position]+"");
    }else{
        vh.edittext.setText("");
    }

    return convertView;
}
```

Adaptor Design

by Wang Duo

Custom Adaptor

-- Effect

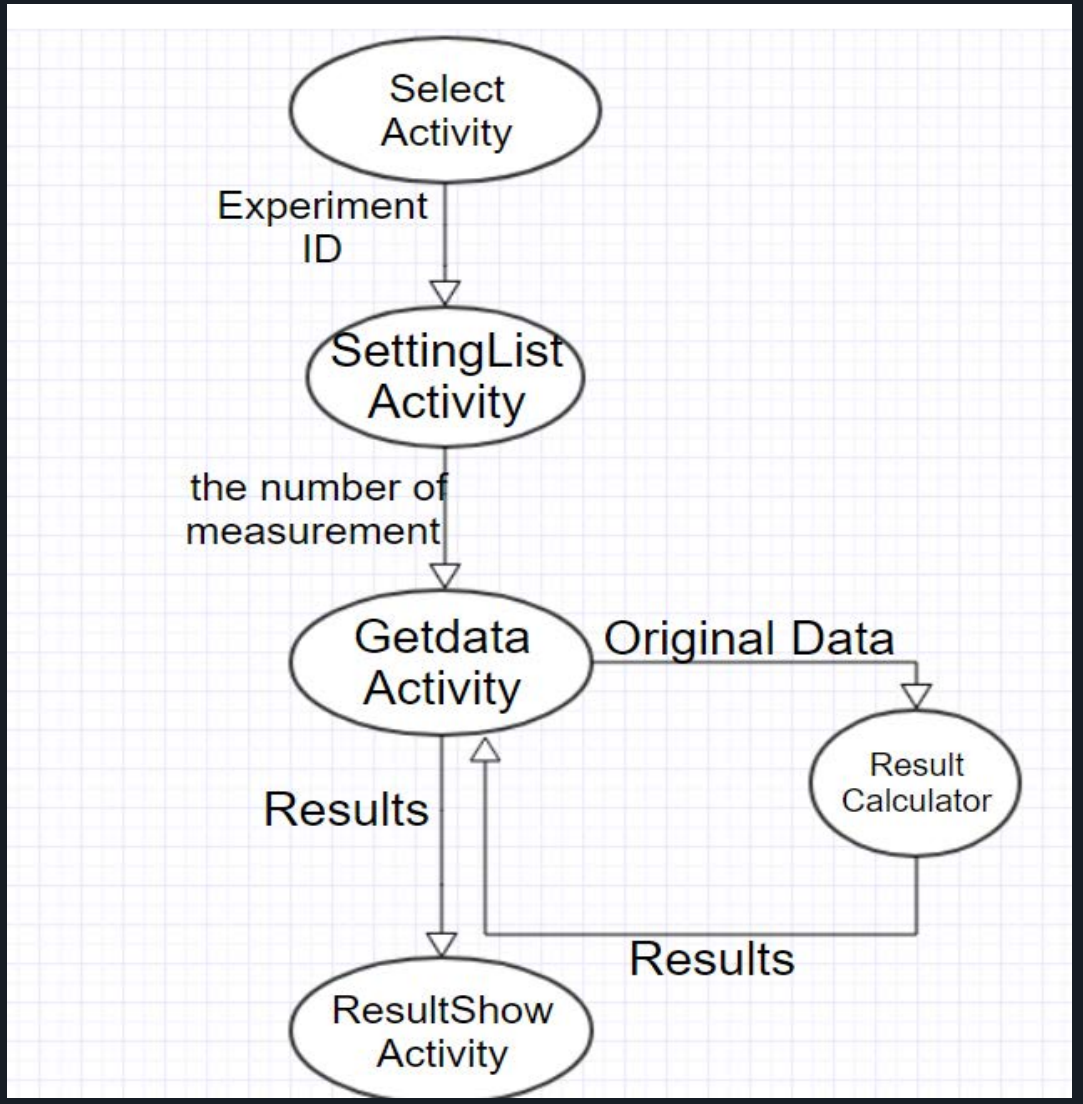
by Jia Xiaosong

Data Acquisition and Transmission 4

by Jia Xiaosong

Architecture of data transmission

-- Select Activity
-- SettingList Activity
-- Getdata Activity
-- Result Calculator
-- Resultshow Activty



by Jia Xiaosong

Realization of Data Acquisition and Transmission

-- intent

```
Intent intent = getIntent();
datas = intent.getIntArrayExtra("MEASUREMENT");
size = intent.getIntExtra("SIZE", 0);
String[] names = intent.getStringArrayExtra("NAMES");
result_id = intent.getIntExtra("SELECTION", 0);
```

-- key-value pair

```
Intent intent = new Intent();
intent.putExtra("RESULTS", results);
intent.putExtra("RESULTNAME", result_name);
intent.putExtra("RESULTNUMBER",result_number);
intent.setClass(get_data.this, ShowActivity.class);
startActivity(intent);
```

by Jia Xiaosong

Data process before passing

-- dynamic create the view

```
int total = 0;
for (int i = 0; i < size; ++i)
    total += datas[i];
ArrayList<HashMap<String, String>> list = new ArrayList<~>();
HashMap<String, String>[] maps = new HashMap[total];

int counter = 0;
for (int i = 0; i < size; i++)
    for (int j = 0; j < datas[i]; ++j) {
        maps[counter] = new HashMap<String, String>();
        if (j == 0)
            maps[counter].put("variable_name", names[i]);
        else
            maps[counter].put("varible_name", "        ");
        list.add(maps[counter]);
        counter++;
    }
```

-- get the result

```
switch (result_id) {
    case 0:
        results = Electron_Charge_Mass_Ratio.get_result(data);
        result_name = resname0;
        result_number = result_num[0];
        break;

    case 1:
        results = Moment_of_Inertia.get_result(data);
        result_name = resname1;
        result_number = result_num[1];
```

by Chai Zhenghao

Calculation Algorithm

5

by Chai Zhenghao

Calculation Components Designing

1.Input Designing

Two-dimensional Arrays

2.Processing the data

3.Output Designing

Arrays—multiple data output

by Chai Zhenghao

Input Designing

Using two-dimensional arrays for storage

data[][]

	data[*][0]	data[*][1]	data[*][2]	...
data[0] coil length	0.241	0.239	0.239	...
data[1] external diameter	0.473	0.477		
data[2] internal diameter	0.464	0.466	0.467	
data[3] number of winding	1425			
...				

Using sparse arrays in Java

by Chai Zhenghao

Processing the data

1.Class for Calculation

Focal_Distance

Electron_Charge_Mass_Ratio

Moment_of_Inertia

Optical_Angel_Gauge

Member function

get_result

```
class Focal_Distance
{
    private static double width_diff = 0.6;
    private static double zizhunfa(double data[][])

    {
        double x1 = Math_Cal.average(data[1]);
        double x2 = Math_Cal.average(data[2]);
        return (x1 + x2 + width_diff) / 2 - Math_Cal.average(data[0]);
    }

    private static double gongefa(double data[][])
    {
        double x1 = Math_Cal.average(data[1]);
        double x2 = Math_Cal.average(data[2]);
        double d1 = Math_Cal.average(data[3]) - Math_Cal.average(data[0]);
        double d2 = x2 - x1;
        double result;
        result = (d1 * d1 - d2 * d2) / (4 * d1);
        return result;
    }
```

by Chai Zhenghao

Processing the data

2.Custom Library

Math_Cal

least_square_method

average

significance_digit

Further plan

User-Defined Experiments

```
class Math_Cal{
    private static double average(double arr[]){
        double sum = 0;
        for (double ele:arr)
            sum += ele;
        return sum/arr.length;
    }

    private static double error(double arr[]){
        double aver = Math_Cal.average(arr);
        double sum = 0;
        for (double ele:arr){
            sum += (aver - ele) * (aver - ele);
        }
        return  Math.sqrt(sum / (arr.length - 1));
    }
```

Possible Impovements

6

Possible Improvements

- More Fancy -- Better Visual Effects
- More Efficient -- Better Interface
- More Universality -- Customized Calculator

Q&A

7


Thanks for listening!