

Bài tập 20: Thực hành về DatePickerDialog và TimePickerDialog trong Android

- Bài tập này bạn sẽ được học về **Time Selection**. Có nhiều loại Time Selection: **AnalogClock**, **Chronometer**, **DatePickerDialog**, **TimePickerDialog**. Nhưng trong bài sẽ giới thiệu 2 control datePicker và timePicker:



DatePickerDialog



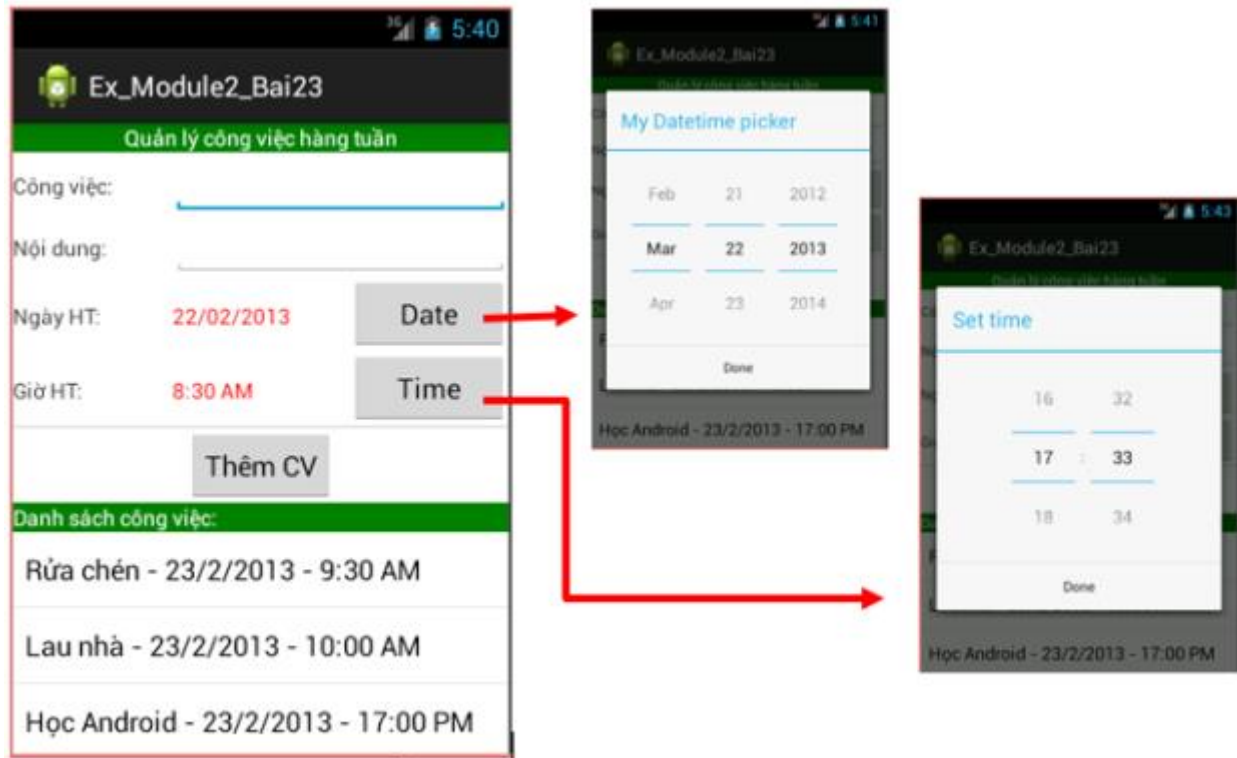
TimePickerDialog

- Hình bên trái là **DatePickerDialog** (dùng để thiết lập ngày tháng năm), hình bên phải là **TimePickerDialog** (dùng để thiết lập giờ phút).

- Không nên kéo thả các control này ra màn hình vì nó chiếm hết không gian của các control khác, mặt khác màn hình điện thoại có không gian hiển thị nhỏ nên nó sẽ không phù hợp khi người học cố tình kéo thả điều khiển ra giao diện. Vì vậy lời khuyên cho bạn là nên viết code động, cho một Button để người sử dụng chọn Ngày tháng hay giờ giấc, khi nhấn vào nó thì mới hiển thị các dialog tương ứng này lên. Ứng dụng thực hành được yêu cầu như sau:

Yêu cầu ứng dụng như sau:

- Viết chương trình quản lý mục tiêu hoàn thành công việc hàng tuần: Lý do chúng ta làm việc không có quy củ, không nhất quán, gặp đâu làm đó, đang làm việc này thấy việc kia xảy ra lại nhảy qua việc đó làm để dở công việc cũ chưa xong, không theo quy trình định sẵn dẫn đến kết thúc một ngày mà cuối cùng không hoàn thành được việc gì cả.
- Thiết kế giao diện như hình bên dưới và thực hiện một số chức năng:



1. Khi chọn nút Date sẽ hiển thị DatePickerDialog – cập nhật Ngày hoàn thành
2. Khi chọn nút Time sẽ hiển thị TimePickerDialog – cập nhật giờ hoàn thành
3. Khi chọn nút Thêm công việc (Thêm CV), chương trình sẽ cập nhật vào ListView bên dưới màn hình
4. Khi nhấn vào phần tử nào thì hiển thị chi tiết **Nội Dung** công việc phải hoàn thành
5. Khi nhấn thật lâu vào phần tử nào trên ListView thì xóa phần tử (công việc) đó.

- Bài tập này đưa ra là tiền đề để làm phần **Android Service** cụ thể là **Alarm Manager**(sinh viên tự tìm hiểu và hoàn thiện bài học).

- Ta sẽ giải quyết yêu cầu bài toán này như sau:

1) Trước tiên Tôi sẽ ôn tập lại **Date**, **Calendar**, **SimpleDateFormat** cho bạn:

- Cách sử dụng Date để xuất Ngày/Tháng / Năm:

```
Public static void xuấtNgàyThangNam()
{
    //Tạo đối tượng date sẽ lấy date hiện tại
    Date date = new Date();
    //Muốn xuất Ngày/Tháng/Năm , ví dụ 12/04/2013

    String strDateFormat = "dd/MM/yyyy";
    //tạo đối tượng SimpleDateFormat;
    SimpleDateFormat sdf = new SimpleDateFormat(strDateFormat);

    //gọi hàm format để lấy chuỗi ngày tháng năm đúng theo yêu cầu
    System.out.println("Ngày hôm nay : "+ sdf.format(date));
}
```

- Tại thời điểm này thì gọi hàm trên sẽ ra kết quả: **Ngày hôm nay : 12/04/2017**

- Cách sử dụng Date để xuất Giờ:Phút:Giây :

```

Public static void xuatGioPhutGiay()
{
    //Tạo đối tượng date sẽ lấy date hiện tại
    Date date = new Date();

    //Muốn xuất Giờ:Phút:Giây AM (PM)
    String strDateFormat12 = "hh:mm:ss a";
    String strDateFormat24 = "HH:mm:ss a";

    SimpleDateFormat sdf =null;
    //Tạo đối tượng SimpleDateFormat với định dạng 12
    sdf= newSimpleDateFormat(strDateFormat12);

    //1. gọi hàm format để lấy giờ:phút:giây loại 12
    System.out.println("Giờ định dạng 12 : "+ sdf.format(date));

    //Tạo đối tượng SimpleDateFormat với định dạng 24
    sdf = newSimpleDateFormat(strDateFormat24);

    //2. gọi hàm format để lấy giờ:phút:giây loại 24
    System.out.println("Giờ định dạng 24 : "+ sdf.format(date));
}

```

- Khi gọi hàm trên thì Tôi sẽ có kết quả (tính tại thời điểm đó):

Giờ định dạng 12 : 04:35:08 PM

Giờ định dạng 24 : 16:35:08 PM

- Bạn quan sát dòng lệnh : “**hh:mm:ss a**”, với **hh** là viết chữ thường thì bạn sẽ được định dạng giờ 12. ký tự **a** ở cuối dùng để hiện thị AM hay là PM tùy thuộc vào giờ lúc đó.

- Bạn quan sát dòng lệnh : “**HH:mm:ss a**”, với **HH** là viết chữ HOA thì bạn sẽ được định dạng giờ 24. ký tự **a** ở cuối dùng để hiện thị AM hay là PM tùy thuộc vào giờ lúc đó. Nhưng bạn nhớ một điều là khi sử dụng định dạng giờ là 24 thì chả ai gán kèm theo AM hay PM đằng sau làm gì. (Không biết Tôi có nói đúng không nhỉ? nếu không đúng thì các bạn cứ cho là Tôi nói đúng để an ủi cũng được) Nên đối với định dạng 24 giờ bạn nên bỏ ký tự **a** đằng sau ra.

- Chú ý lập trình viên có thể kết hợp hiển thị Ngày/Tháng/Năm giờ hiện tại:

“**dd/MM/yyyyhh:mm:ss a**”

- Cách sử dụng Calendar (class này bây giờ rất được ưa chuộng, bạn nên tập làm quen với nó):

```

Public static void sudungCalendar()
{
    //Lấy đối tượng Calendar ra, mặc định ngày hiện tại
    Calendar now = Calendar.getInstance();

    //Muốn xuất Giờ:Phút:Giây AM (PM)
    String strDateFormat12 = "hh:mm:ss a";
    String strDateFormat24 = "HH:mm:ss";
    SimpleDateFormat sdf =null;

    //Tạo đối tượng SimpleDateFormat với định dạng 12
    sdf= new SimpleDateFormat(strDateFormat12);

    //1. gọi hàm format để lấy giờ:phút:giây loại 12
    System.out.println("Giờ định dạng 12 : "+sdf.format(now.getTime()));
    //Tạo đối tượng SimpleDateFormat với định dạng 24

```

```

sdf = new SimpleDateFormat(strDateFormat24);
//2. gọi hàm format để lấy giờ:phút:giây loại 24
System.out.println("Giờ định dạng 24 : "+sdf.format(now.getTime()));

String strDateFormat = "dd/MM/yyyy hh:mm:ss a";//chú ý chữ hoa
sdf = new SimpleDateFormat(strDateFormat);
System.out.println("Bây giờ là: "+ sdf.format(now.getTime()));

System.out.println("Năm hiện tại : "+ now.get(Calendar.YEAR));
//Tháng tính từ 0 tới 11, nên phải + thêm 1
System.out.println("Tháng hiện tại : "+(now.get(Calendar.MONTH) + 1));
System.out.println("Ngày hiện tại : "+now.get(Calendar.DATE));

System.out.println("Lấy giờ định dạng 12:" + now.get(Calendar.HOUR));
System.out.println("Lấy giờ định dạng 24:" + now.get(Calendar.HOUR_OF_DAY));
System.out.println("Phút hiện tại : "+now.get(Calendar.MINUTE));
System.out.println("Giây hiện tại : "+now.get(Calendar.SECOND));
System.out.println("Mili giây hiện tại: "+now.get(Calendar.MILLISECOND));
}

```

- Nhìn vào giải thích ở trên thì ta thấy khi muốn lấy Năm, Tháng, Ngày, giờ ,Phút , giây thì phải sử dụng hằng số nào của Calendar.

- Thực hiện chương trình trên thì bạn sẽ có kết quả như bên dưới (tính tại thời điểm ví dụ):

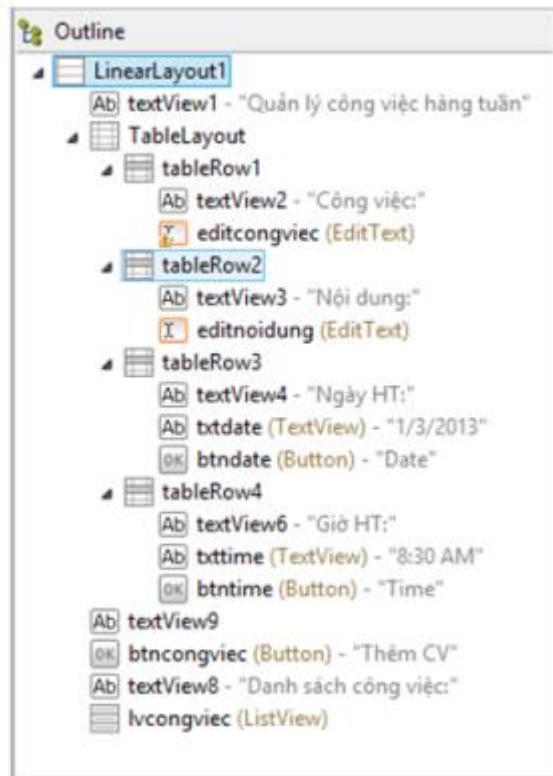
```

Giờ định dạng 12 : 05:02:50 PM
Giờ định dạng 24 : 17:02:50
Bây giờ là: 12/04/2013 05:02:50 PM
Năm hiện tại : 2013
Tháng hiện tại : 4
Ngày hiện tại : 12
Lấy giờ định dạng 12 là : 5
Lấy giờ định dạng 24 là : 17
Phút hiện tại : 2
Giây hiện tại : 50
Mili giây hiện tại: 527

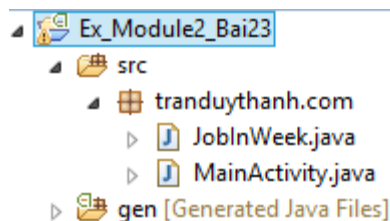
```

- 2) – Phần trên đã trình bày cách sử dụng Date, Calendar, SimpleDateFormat, Giờ bắt đầu làm ứng dụng:

- Tạo Android Project tên tùy thích và thiết kế giao diện giống như yêu cầu bên trên:



- Sau đây là nội dung xử lý các nghiệp vụ:



- Class **JobInWeek** để định nghĩa một công việc phải hoàn thành trong một tuần.

```

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
/**
 * Class được định nghĩa công việc phải hoàn thành
 * trong một tuần
 *
 */
Public class JobInWeek {
    Private String title;
    Private String description;
    Private Date dateFinish;
    Private Date hourFinish;
    //xây dựng các hàm get/set các thuộc tính
    //xây dựng hàm tạo phù hợp

    /**
     * lấy định dạng ngày
     * @param d
     * @return
     */
    PublicString getDateFormat(Date d)
    {
        SimpleDateFormat dft=new

```

```

        SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());
        returndft.format(d);
    }
    /**
    * lấy định dạng giờ phút
    * @param d
    * @return
    */
    Public String getHourFormat(Date d)
    {
        SimpleDateFormat dft=new
        SimpleDateFormat("hh:mm a", Locale.getDefault());
        returndft.format(d);
    }
    @Override
    Public String toString() {
        returnthis.title+"-"+
        getDateFormat(this.dateFinish)+"-"+
        getHourFormat(this.hourFinish);
    }
}

```

- Class MainActivity:

```

Public class MainActivity extends Activity {
    TextView txtDate,txtTime;
    EditText editCv,editNd;
    Button btnDate,btnTime,btnAdd;
    //Khai báo Datasource lưu trữ danh sách công việc
    ArrayList<JobInWeek> arrJob=new ArrayList<JobInWeek>();
    //Khai báo ArrayAdapter cho ListView
    ArrayAdapter<JobInWeek> adapter=null;
    ListView lvCv;
    Calendar cal;
    Date dateFinish;
    Date hourFinish;
    @Override
    Protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getFormWidgets();
        getDefaultInfor();
        addEventFormWidgets();
    }
    /**
    * hàm dùng để load các control theo Id
    */
    Public void getFormWidgets()
    {
        txtDate=(TextView) findViewById(R.id.txtdate);
        txtTime=(TextView) findViewById(R.id.txttime);
        editCv=(EditText) findViewById(R.id.editcongviiec);
        editNd=(EditText) findViewById(R.id.editnoidung);
        btnDate=(Button) findViewById(R.id.btndate);
        btnTime=(Button) findViewById(R.id.btntime);
        btnAdd=(Button) findViewById(R.id.btncongviiec);
        lvCv=(ListView) findViewById(R.id.lvcongviiec);
        //Gán DataSource vào ArrayAdapter
        adapter=new ArrayAdapter<JobInWeek>(
            this,
            android.R.layout.simple_list_item_1,
            arrJob);
        //gán Adapter vào ListView
        lvCv.setAdapter(adapter);
    }
}

```

```

}
/**
 * Hàm lấy các thông số mặc định khi lần đầu tiên chạy ứng dụng
 */
Public void getDefaultInfor()
{
    //lấy ngày hiện tại của hệ thống
    cal=Calendar.getInstance();
    SimpleDateFormat dft=null;

    //Định dạng ngày / tháng /năm
    dft=new SimpleDateFormat("dd/MM/yyyy",Locale.getDefault());
    String strDate=dft.format(cal.getTime());

    //hiển thị lên giao diện
    txtDate.setText(strDate);

    //Định dạng giờ phút am/pm
    dft=new SimpleDateFormat("hh:mm a",Locale.getDefault());
    String strTime=dft.format(cal.getTime());

    //đưa lên giao diện
    txtTime.setText(strTime);
    //lấy giờ theo 24 để lập trình theo Tag
    dft=new SimpleDateFormat("HH:mm",Locale.getDefault());
    txtTime.setTag(dft.format(cal.getTime()));

    editCv.requestFocus();
    //gán cal.getTime() cho ngày hoàn thành và giờ hoàn thành
    dateFinish=cal.getTime();
    hourFinish=cal.getTime();
}
/**
 * Hàm gán các sự kiện cho các control
 */
Public void addEventFormWidgets()
{
    btnDate.setOnClickListener(new MyButtonEvent());
    btnTime.setOnClickListener(new MyButtonEvent());
    btnAdd.setOnClickListener(new MyButtonEvent());
    lvCv.setOnItemClickListener(new MyListViewEvent());
    lvCv.setOnItemLongClickListener(new MyListViewEvent());
}
/**
 * Class sự kiện của các Button
 * @author drthanh
 */
Private class MyButtonEvent implements OnClickListener
{
    @Override
    Public void onClick(View v) {
        switch(v.getId())
        {
            Case R.id.btndate:
                showDatePickerDialog();
                break;
            case R.id.btntime:
                showTimePickerDialog();
                break;
            case R.id.btncongviec:
                processAddJob();
                break;
        }
    }
}

```

```

}

}
/**
 * Class sự kiện của ListView
 * @author drthanh
 *
 */
Private class MyListViewEvent implements OnItemClickListener,
OnItemLongClickListener
{

@Override
Public boolean onItemLongClick(AdapterView<?> arg0, View arg1, intarg2,
    longarg3) {
//Xóa vị trí thứ arg2
    arrJob.remove(arg2);
    adapter.notifyDataSetChanged();
    returnfalse;
}

@Override
Public void onItemClick(AdapterView<?> arg0, View arg1, intarg2,
    longarg3) {
    //Hiển thị nội dung công việc tại vị trí thứ arg2
    Toast.makeText(MainActivity.this,
        arrJob.get(arg2).getDescription(),
        Toast.LENGTH_LONG).show();
}

}
/**
 * Hàm hiển thị DatePicker dialog
 */
Public void showDatePickerDialog() {
    OnDateSetListener callback=new OnDateSetListener() {
        PublicvoidonDateSet(DatePicker view, intyear, intmonthOfYear,
            intdayOfMonth) {
            //Mỗi lần thay đổi ngày tháng năm thì cập nhật lại TextView Date
            txtDate.setText( (dayOfMonth) + "/" + (monthOfYear+1) + "/" + year);
            //Lưu vết lại biến ngày hoàn thành
            cal.set(year, monthOfYear, dayOfMonth);
            dateFinish=cal.getTime();
        }
    };
    //các lệnh dưới này xử lý ngày giờ trong DatePickerDialog
    //sẽ giống với trên TextView khi mở nó lên
    String s=txtDate.getText()+" ";
    String strArrtmp[]=s.split("/");
    intngay=Integer.parseInt(strArrtmp[0]);
    intthang=Integer.parseInt(strArrtmp[1])-1;
    intnam=Integer.parseInt(strArrtmp[2]);

    DatePickerDialog pic=newDatePickerDialog(
        MainActivity.this,
        callback, nam, thang, ngay);
    pic.setTitle("Chọn ngày hoàn thành");
    pic.show();
}

```



```

/**
 * Hàm hiển thị TimePickerDialog
 */
Public void showTimePickerDialog()
{
    OnTimeSetListener callback=new OnTimeSetListener() {
        Public void onTimeSet(TimePicker view, InthourOfDay, intminute) {
            //Xử lý lưu giờ và AM, PM
            String s=hourOfDay + ":" +minute;
            inthourTam=hourOfDay;
            if(hourTam>12)
                hourTam=hourTam-12;
            txtTime.setTexthourTam + ":" +minute + (hourOfDay>12?" PM":" AM"));
            //lưu giờ thực vào tag
            txtTime.setTag(s);
            //lưu vết lại giờ vào hourFinish
            cal.set(Calendar.HOUR_OF_DAY, hourOfDay);
            cal.set(Calendar.MINUTE, minute);
            hourFinish=cal.getTime();
        }
    };
    //các lệnh dưới này xử lý ngày giờ trong TimePickerDialog
    //sẽ giống với trên TextView khi mở nó lên
    String s=txtTime.getTag()+" ";
    String strArr[]=s.split(":");
    intgio=Integer.parseInt(strArr[0]);
    intphut=Integer.parseInt(strArr[1]);
    TimePickerDialog time=new TimePickerDialog(
        MainActivity.this,
        callback, gio, phut, true);
    time.setTitle("Chọn giờ hoàn thành");
    time.show();
}
/**
 * Hàm xử lý đưa công việc vào ListView khi nhấn nút Thêm Công việc
 */
Public void processAddJob()
{
    String title=editCv.getText()+" ";
    String description=editNd.getText()+" ";
    JobInWeek job=new JobInWeek(title, description, dateFinish, hourFinish);
    arrJob.add(job);
    adapter.notifyDataSetChanged();
    //sau khi cập nhật thì reset dữ liệu và cho focus tới editCV
    editCv.setText("");
    editNd.setText("");
    editCv.requestFocus();
}
}

```

- Như đã nói ở các bài thực hành trước là chúng ta nên tách các hàm ra để dễ xử lý, thì trong bài trên cũng vậy. Tôi cũng đã ghi chú trực tiếp vào coding, càng về sau Tôi sẽ ghi chú ít đi, chỉ ghi chú những cái gì mới mà thôi.

- Trong đoạn code trên ta sẽ sử dụng 2 call back listener- mục đích là lắng nghe sự thay đổi dữ liệu trong Các dialog tương ứng:

+ Interface **OnDataSetListener** có phương thức trừa hưởng **onDataSet**, trong phương thức này ta xử lý sự thay đổi dữ liệu trong **DatePickerDialog**.

+ Interface **OnTimeSetListener** có phương thức trừu tượng **onTimeSet**, trong phương thức này ta xử lý sự thay đổi giờ – phút trong **TimePickerDialog**.

- 2 call back listener trên rất là hữu dụng, dựa vào nó mà ta luôn kiểm tra được sự thay đổi dữ liệu trong các Dialog.

- Bài tập kế tiếp bạn sẽ học về **Tab Selector (cũ)**, một control dùng để chia màn hình thành các phần tương tác khác nhau, nó cũng thú vị không kém. Hoặc bạn có thể tìm hiểu control mới là **ActionBar** trong Android 3.0 (API 11). Chúc bạn thành công.