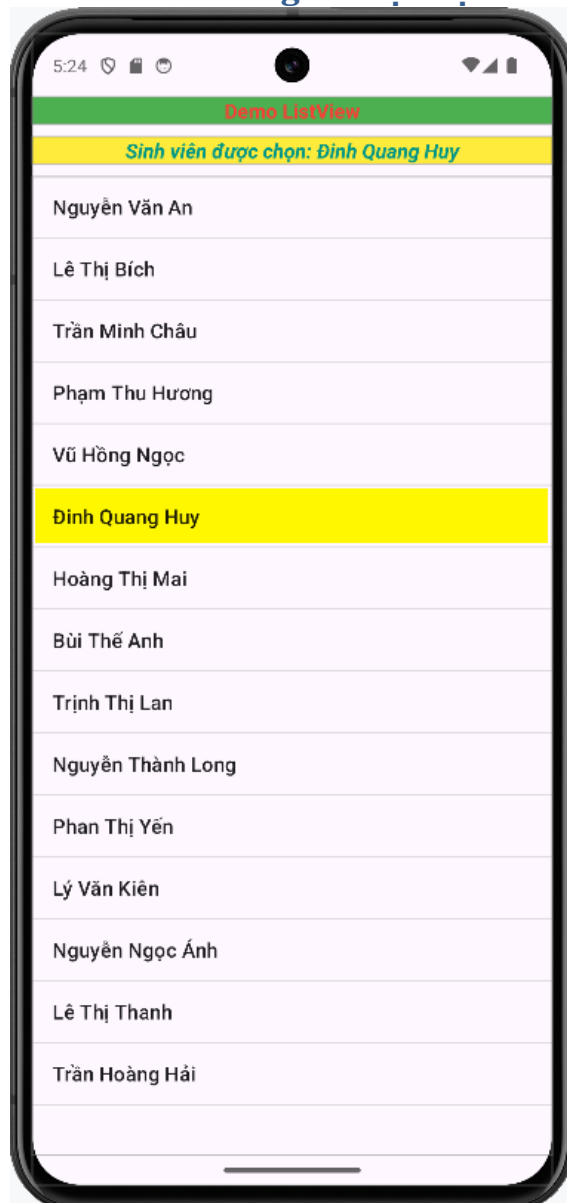
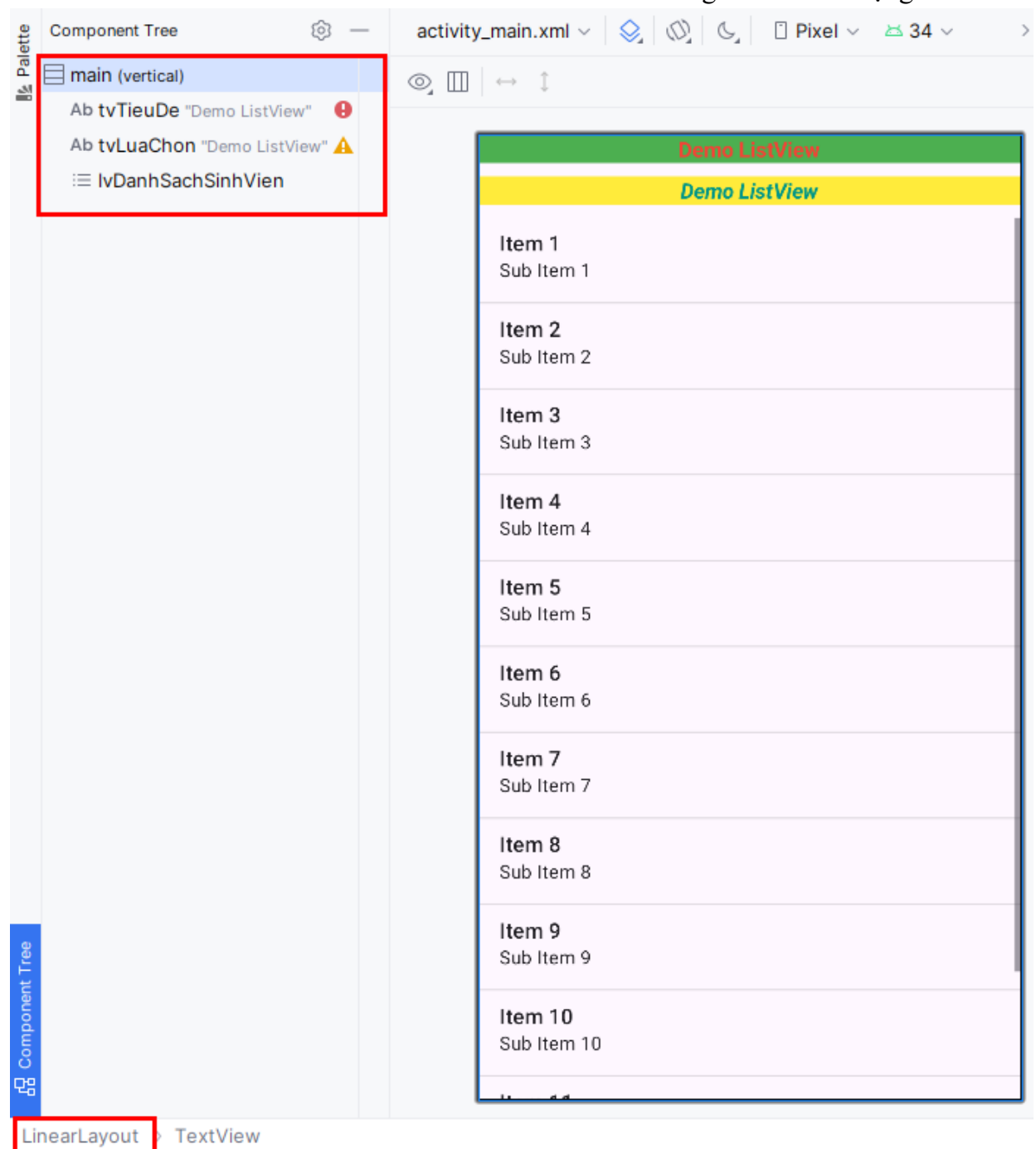


Bài 1. Sử dụng ListView control với mảng dữ liệu định sẵn

- ❖ **Yêu cầu:** Hiển thị các phần tử trong mảng dữ liệu có sẵn lên ListView
- ❖ **Hướng dẫn:**
 - File giao diện `activity_main.xml`:



- **File code MainActivity.java:**

```
public class MainActivity extends AppCompatActivity {
```

```
    ListView lvDanhSachSinhVien; 3 usages
    TextView tvLuaChon; 2 usages
    String[] students = { 2 usages
        "Nguyễn Văn An", "Lê Thị Bích", "Trần Minh Châu",
        "Phạm Thu Hương", "Vũ Hồng Ngọc", "Đinh Quang Huy",
        "Hoàng Thị Mai", "Bùi Thế Anh", "Trịnh Thị Lan",
        "Nguyễn Thành Long", "Phan Thị Yến", "Lý Văn Kiên",
        "Nguyễn Ngọc Ánh", "Lê Thị Thanh", "Trần Hoàng Hải"
    };
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
}
```

```
// Bước 1: Ánh xạ các thành phần giao diện
```

```
lvDanhSachSinhVien = findViewById(R.id.lvDanhSachSinhVien);
tvLuaChon = findViewById(R.id.tvLuaChon);
```

```
// Bước 2: Gán datasource (mảng students) vào ArrayAdapter
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<>(context, this, android.R.layout.simple_list_item_1, students);
```

```
// Bước 3: Gán adapter vào ListView
```

```
lvDanhSachSinhVien.setAdapter(adapter);
```

```
// Bước 4: Xử lý sự kiện khi item trong ListView được chọn
```

```
lvDanhSachSinhVien.setOnItemClickListener((parent, view, position, id) -> {
    tvLuaChon.setText("Sinh viên được chọn: " + students[position]);
});
```

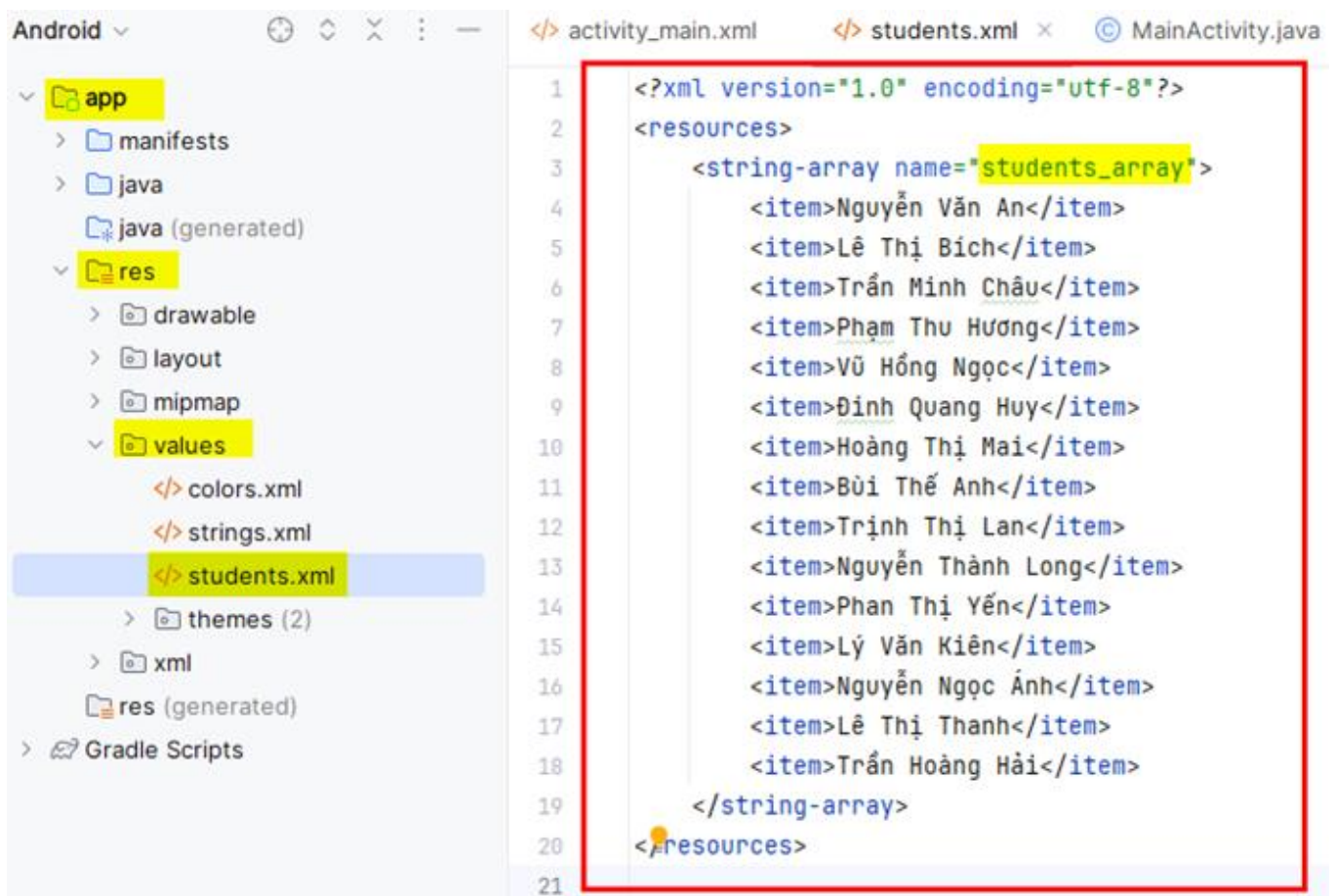
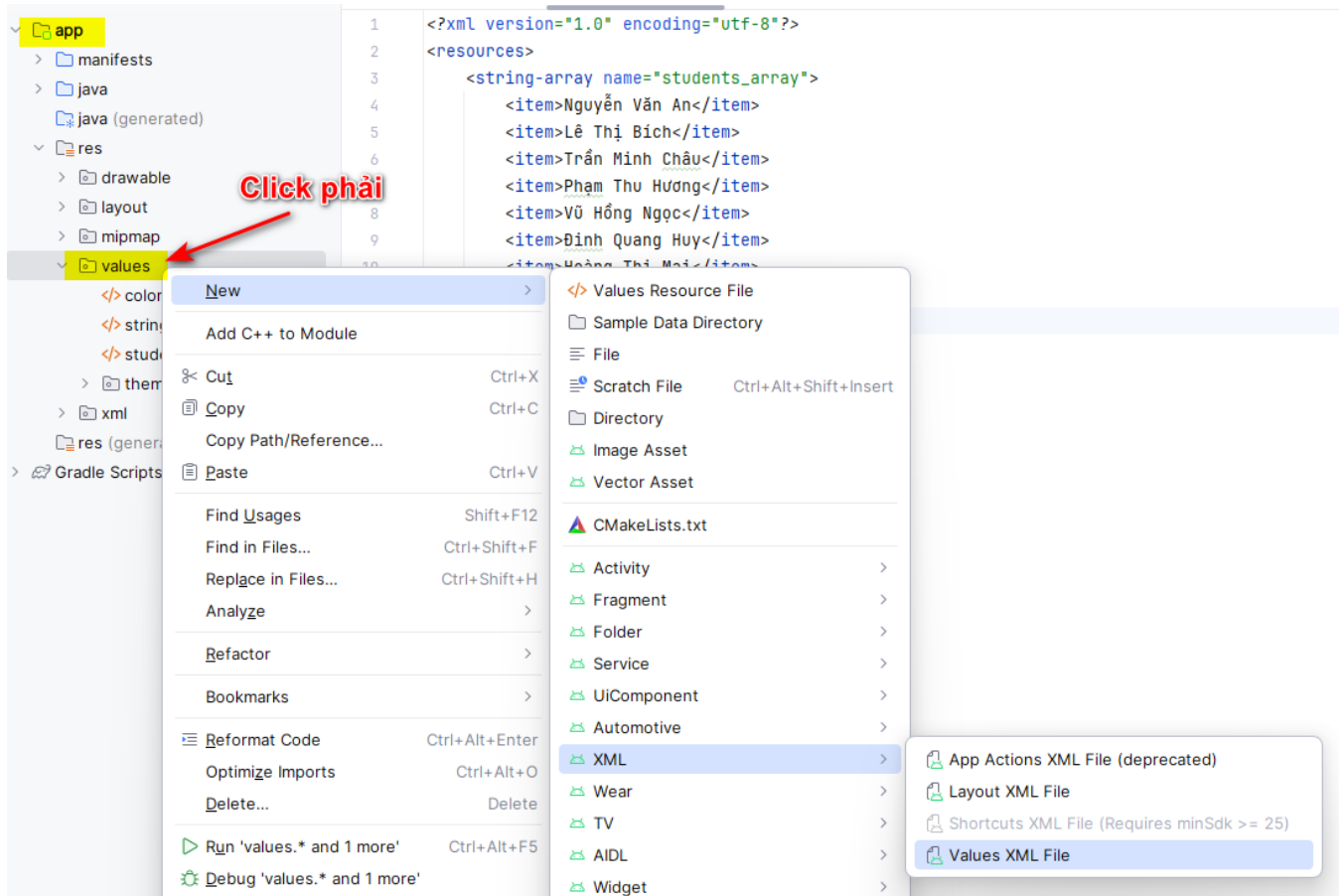
```
}
```

```
}
```

Bài 2. Sử dụng ListView với mảng dữ liệu được lưu trong Xml

- ❖ **Yêu cầu:** Tạo xml file chứa danh sách Sinh viên và ứng dụng sẽ load nội dung xml này lên ListView
- ❖ **Hướng dẫn:**

- Tạo students.xml:

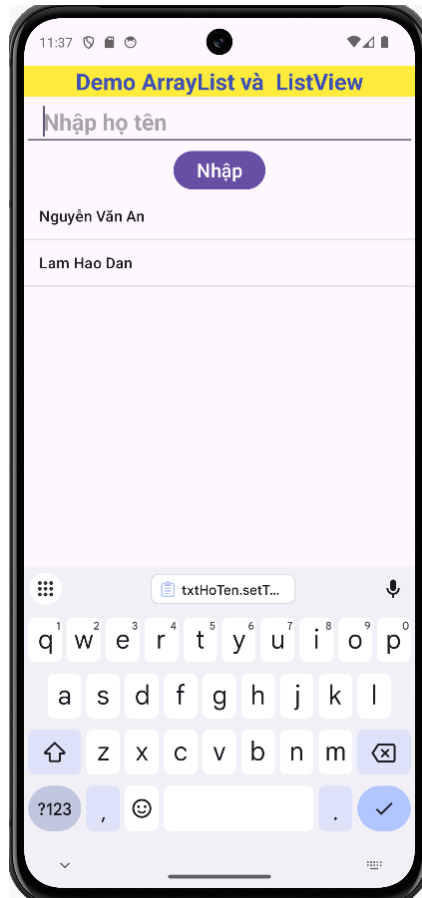


- Khởi tạo mảng **students** bằng cách load dữ liệu từ resource xml:

```
public class lab05_bai02 extends AppCompatActivity {

    ListView lvDanhSachSinhVien; 3 usages
    TextView tvLuaChon; 2 usages
    String[] students = getResources().getStringArray(R.array.students_array); 2 usages
}
```

Bài 3. Sử dụng ArrayList và ListView control

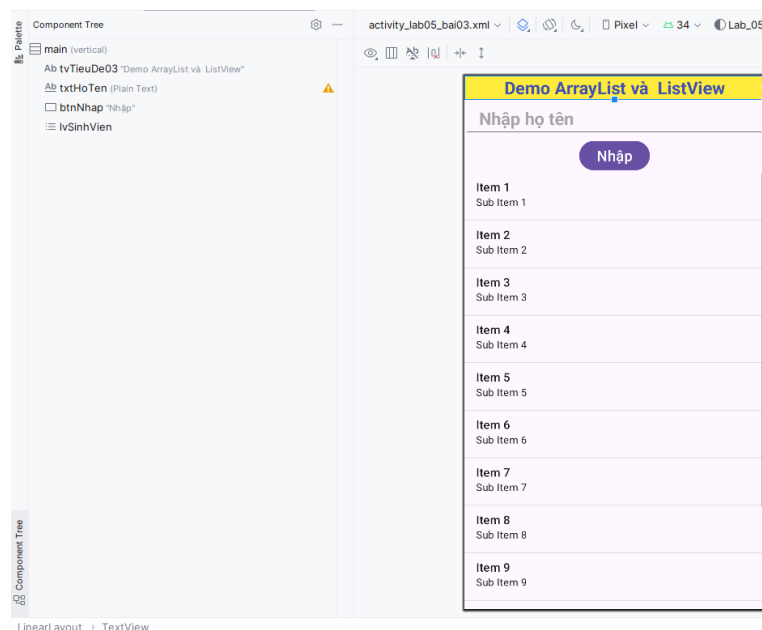


❖ Yêu cầu:

- Người dùng nhập tên sinh viên vào ô “Nhập họ tên sinh viên” và sau đó nhấn nút “Nhập” thì nội dung của EditText sẽ được di chuyển vào trong ListView.
- Khi người dùng nhấn và giữ lâu 1 item trên ListView thì item đó sẽ bị **xóa** khỏi danh sách

❖ Hướng dẫn:

- Thiết kế UI:



- Java code:

```
public class lab05_bai03 extends AppCompatActivity {

    //Bước 0: Khai báo các thành phần giao diện
    ListView lvDanhSachSinhVien; 4 usages
    TextView txtHoTen; 5 usages
    Button btnNhap; 2 usages

    ArrayList<String> dsSinhVien = null; 5 usages
    ArrayAdapter<String> adapterSinhVien = null; 4 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_lab05_bai03);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        //Bước 1: Ảnh xạ, kết nối các thành phần giao diện
        lvDanhSachSinhVien = (ListView) findViewById(R.id.lvSinhVien);
        txtHoTen = (EditText) findViewById(R.id.txtHoTen);
        btnNhap = (Button) findViewById(R.id.btnNhap);
        txtHoTen.setText("Nguyễn Văn An");

        //Bước 2: Khởi tạo ArrayList
        dsSinhVien = new ArrayList<>();

        //Bước 3: Khởi tạo ArrayAdapter
        adapterSinhVien = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, dsSinhVien);

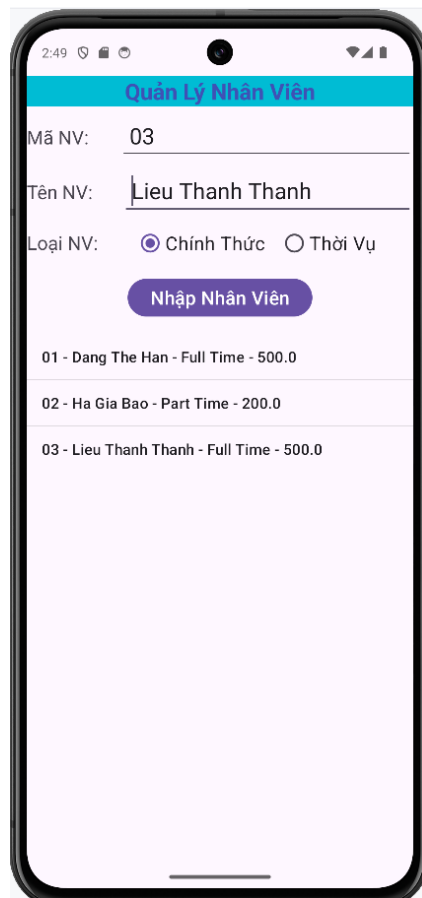
        //Bước 4: Gán adapter vào ListView
        lvDanhSachSinhVien.setAdapter(adapterSinhVien);

        //Bước 5: Xử lý sự kiện khi nút "Nhap" được nhấn
        btnNhap.setOnClickListener(v -> {
            String hoTen = txtHoTen.getText().toString(); // Lấy họ tên từ EditText
            if (!hoTen.isEmpty()) { // Kiểm tra họ tên không rỗng
                dsSinhVien.add(hoTen); // Thêm họ tên mới vào danh sách sinh viên
                adapterSinhVien.notifyDataSetChanged(); // Thông báo cho adapter dữ liệu đã thay đổi, để hiển thị dữ liệu mới lên ListView
                txtHoTen.setText(""); // Xóa nội dung trong EditText
            }
        });

        //Bước 6: Xử lý sự kiện khi item trong ListView được chọn
        lvDanhSachSinhVien.setOnItemClickListener((parent, view, position, id) -> {
            String hoTen = dsSinhVien.get(position); // Lấy họ tên của sinh viên được chọn từ vị trí position
            txtHoTen.setText(hoTen); // Hiển thị lại họ tên lên EditText
        });

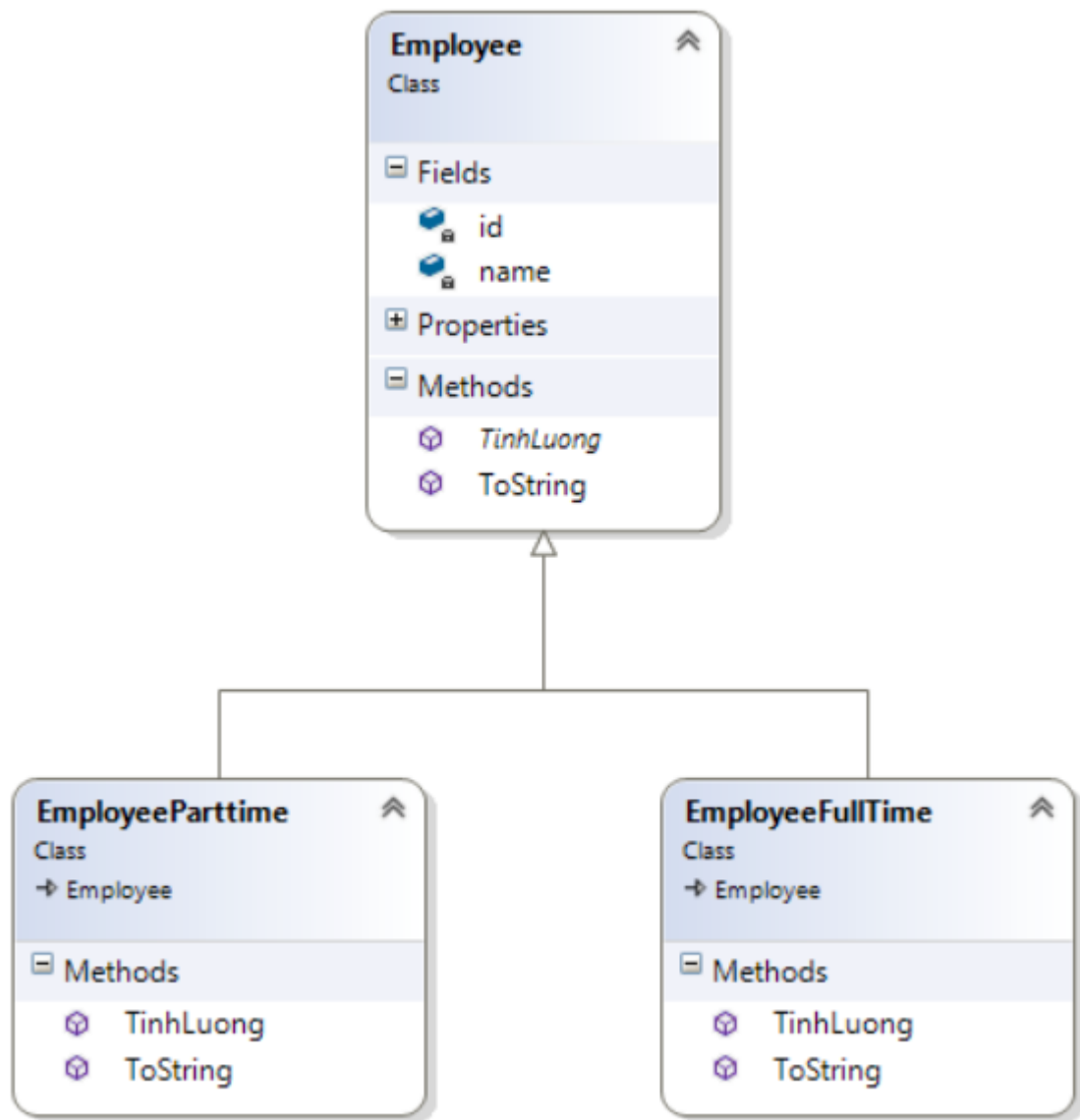
        //Bước 7: Xử lý sự kiện khi item trong ListView được giữ => Nhấn và giữ lại => để xóa
        lvDanhSachSinhVien.setOnItemLongClickListener((parent, view, position, id) -> {
            dsSinhVien.remove(position); // Xóa sinh viên khỏi danh sách
            adapterSinhVien.notifyDataSetChanged(); // Thông báo cho adapter dữ liệu đã thay đổi, để hiển thị dữ liệu mới lên ListView
            return true; // Báo rằng sự kiện long-click đã được xử lý hoàn toàn và không cần xử lý thêm.
        });
    }
}
```

Bài 4. Sử dụng ArrayList và ListView nhưng từng phần tử trong ArrayList là các Object bất kỳ



❖ Yêu cầu:

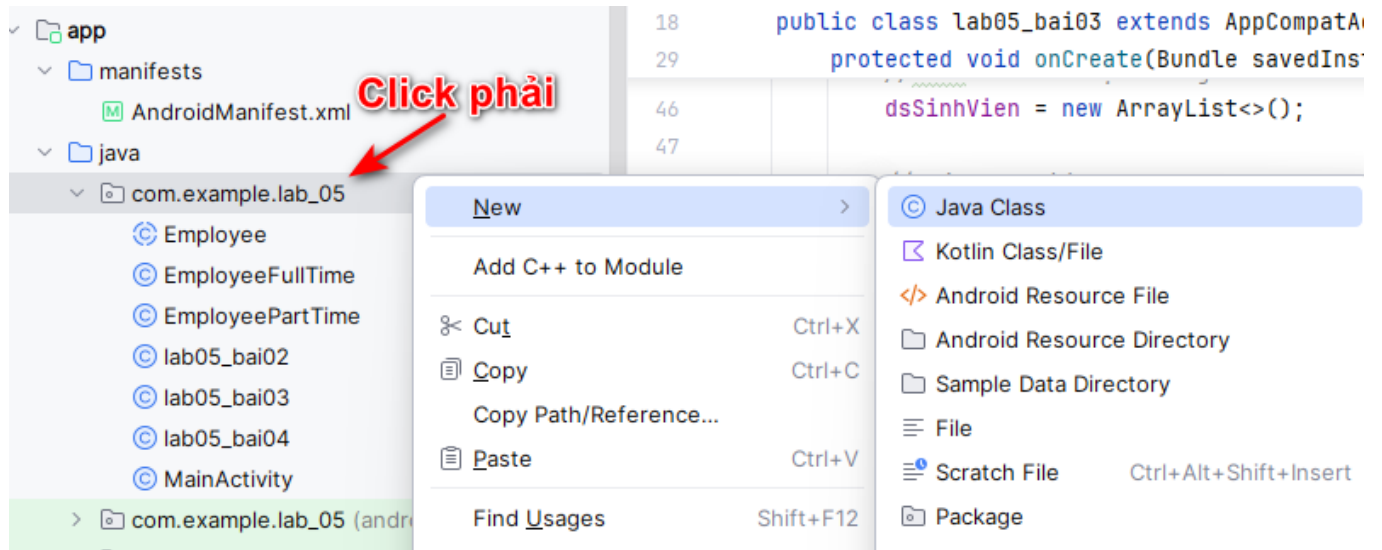
- Người dùng sau khi nhập Mã NV, Tên NV, loại NV thì nhấn nút Nhập Nhân Viên để thêm thông tin nhân viên vừa nhập vào danh sách nhân viên (ListView)
- Người dùng chọn nhân viên trên listview thì thông tin chi tiết sẽ hiển thị ngược trở lại trên editText Mã NV, và editText Tên NV.
- Người dùng có thể chọn và giữ một nhân viên trong danh sách để xóa
- Cho sơ đồ class như sau:



- Có 2 loại nhân viên :
 - Nhân viên chính thức (EmployeeFullTime)
 - Nhân viên thời vụ (EmployeeParttime)
- Mỗi nhân viên sẽ có cách tính lương khác nhau (tên phương thức tính lương giống nhau: **TinhLuong()**)
 - Lương nhân viên chính thức là: 500
 - Lương nhân viên thời vụ là: 200
- Mỗi nhân viên có phương thức **toString()** để xuất thông tin cho nhân viên đó. Nội dung xuất khác nhau:
 - Thêm "FullTime" đằng sau Id và Name đối với nhân viên chính thức.
 - Thêm "Parttime" đằng sau Id và Name đối với nhân viên thời vụ.

❖ **Hướng dẫn:**

- Employee: Tạo class Employee.java để tiện xử lý



```
public abstract class Employee { 7 usages 2 inheritors
    private String id; 4 usages
    private String name; 4 usages

    public Employee() { 2 usages
    }

    public Employee(String id, String name) { no usages
        this.id = id;
        this.name = name;
    }

    public String getId() { 1 usage
        return id;
    }

    public void setId(String id) { 1 usage
        this.id = id;
    }

    public String getName() { 1 usage
        return name;
    }

    public void setName(String name) { 1 usage
        this.name = name;
    }

    public abstract double tinhLuong(); 2 usages 2 implementations

    @Override 2 overrides
    public String toString() {
        return this.id + " - " + this.name;
    }
}
```

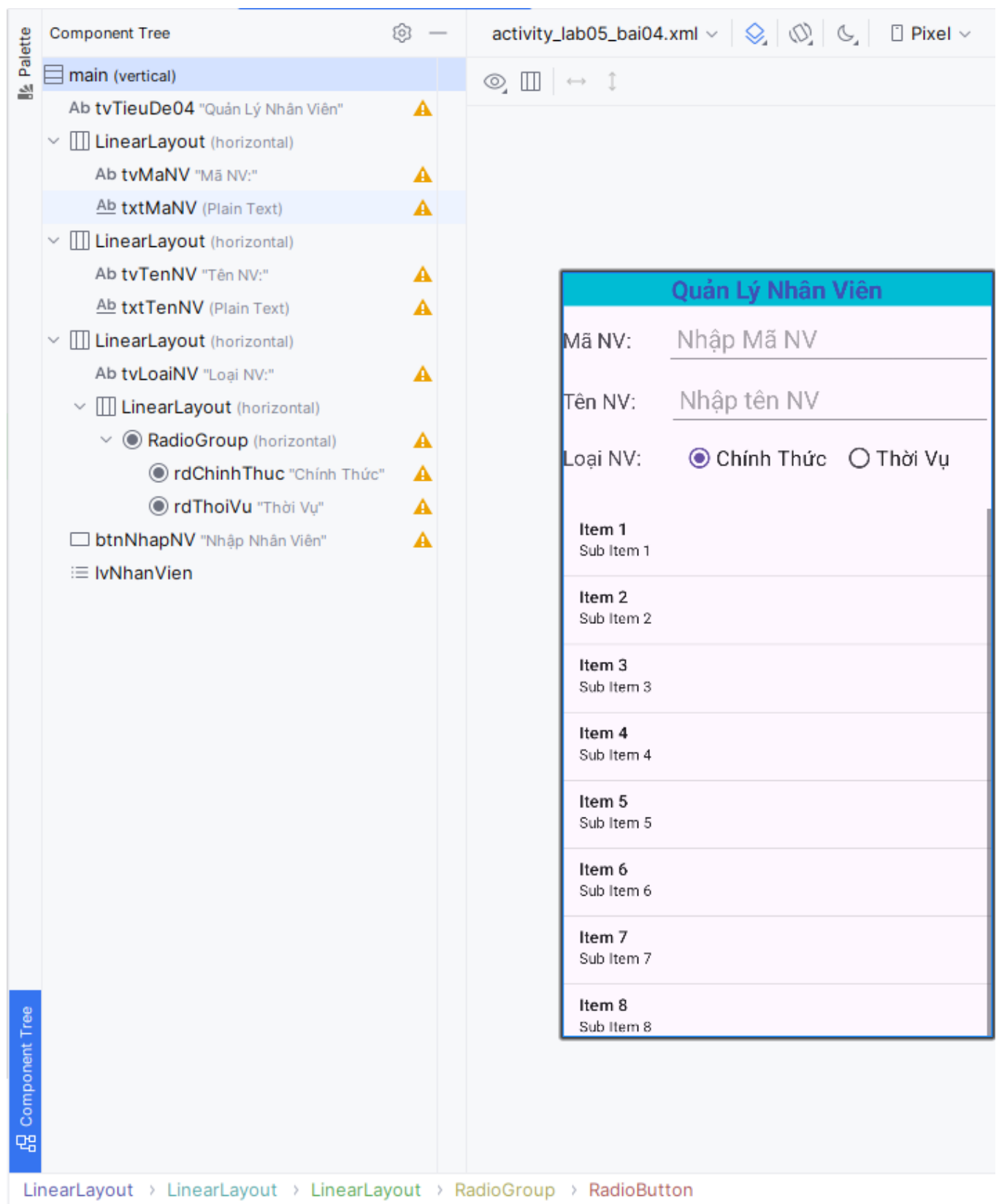
➤ EmployeeFullTime

```
public class EmployeeFullTime extends Employee { 1 usage
    @Override 2 usages
    public double tinhLuong() {
        return 500;
    }

    @Override
    public String toString() {
        return super.toString() + " - Full Time - " + tinhLuong();
    }
}
```

➤ EmployeePartTime: (tương tự EmployeeFullTime)

➤ Giao diện



➤ Java code chính

```
18 public class lab05_bai04 extends AppCompatActivity {
19     //Bước 0: Khai báo các biến
20     Button btnNhap; 2 usages
21     EditText txtMaNV, txtTenNV; 4 usages
22     RadioButton rdChinhThuc, rdThoiVu; 3 usages
23     ListView lvNhanVien; 4 usages
24
25     ArrayList<Employee> employeeArrayList = null; 5 usages
26     ArrayAdapter<Employee> employeeArrayAdapter = null; 4 usages
27     Employee employee = null; 8 usages
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
33         setContentView(R.layout.activity_lab05_bai04);
34         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
35             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
36             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
37             return insets;
38         });
39
40         //Bước 1: Khởi tạo các biến và kết nối các thành phần giao diện
41         btnNhap = (Button) findViewById(R.id.btnNhapNV);
42         txtMaNV = (EditText) findViewById(R.id.txtMaNV);
43         txtTenNV = (EditText) findViewById(R.id.txtTenNV);
44         rdChinhThuc = (RadioButton) findViewById(R.id.rdChinhThuc);
45         rdThoiVu = (RadioButton) findViewById(R.id.rdThoiVu);
46         lvNhanVien = (ListView) findViewById(R.id.lvNhanVien);
47
48         //Bước 2: Khởi tạo danh sách nhân viên
49         employeeArrayList = new ArrayList<Employee>();
50         employeeArrayAdapter = new ArrayAdapter<Employee>( context: this, android.R.layout.simple_list_item_1, employeeArrayList);
51         lvNhanVien.setAdapter(employeeArrayAdapter);
52
53         //Bước 3: Thiết lập sự kiện cho nút nhập
54         btnNhap.setOnClickListener(v -> {
55             //Bước 3.1: Lấy dữ liệu từ các thành phần giao diện
56             String maNV = txtMaNV.getText().toString();
57             String tenNV = txtTenNV.getText().toString();
58             if (rdChinhThuc.isChecked()) {
59                 employee = new EmployeeFullTime();
60             } else {
61                 employee = new EmployeePartTime();
62             }
63             employee.setId(maNV);
64             employee.setName(tenNV);
65             //Bước 3.2: Thêm nhân viên vào danh sách
66             employeeArrayList.add(employee);
67             //Bước 3.3: Cập nhật danh sách nhân viên trên giao diện
68             employeeArrayAdapter.notifyDataSetChanged();
69             //Bước 3.4: Xóa dữ liệu trong các thành phần giao diện
70             txtMaNV.setText("");
71             txtTenNV.setText("");
72             rdChinhThuc.setChecked(true); // Đặt chế độ mặc định là chính thức
73         });
74
75         //Bước 4: Thiết lập sự kiện cho ListView
76         lvNhanVien.setOnItemClickListener((parent, view, position, id) -> {
77             employee = employeeArrayList.get(position);
78             txtMaNV.setText(employee.getId());
79             txtTenNV.setText(employee.getName());
80         });
81
82         //Bước 5: Thiết lập sự kiện chọn và giữ lâu item để xóa một nhân viên
83         lvNhanVien.setOnItemLongClickListener((parent, view, position, id) -> {
84             employeeArrayList.remove(position);
85             employeeArrayAdapter.notifyDataSetChanged();
86             return true;
87         });
88     }
89 }
```

(*) Hướng dẫn cài lombok**Bước 1: Cài đặt Plugin Lombok**

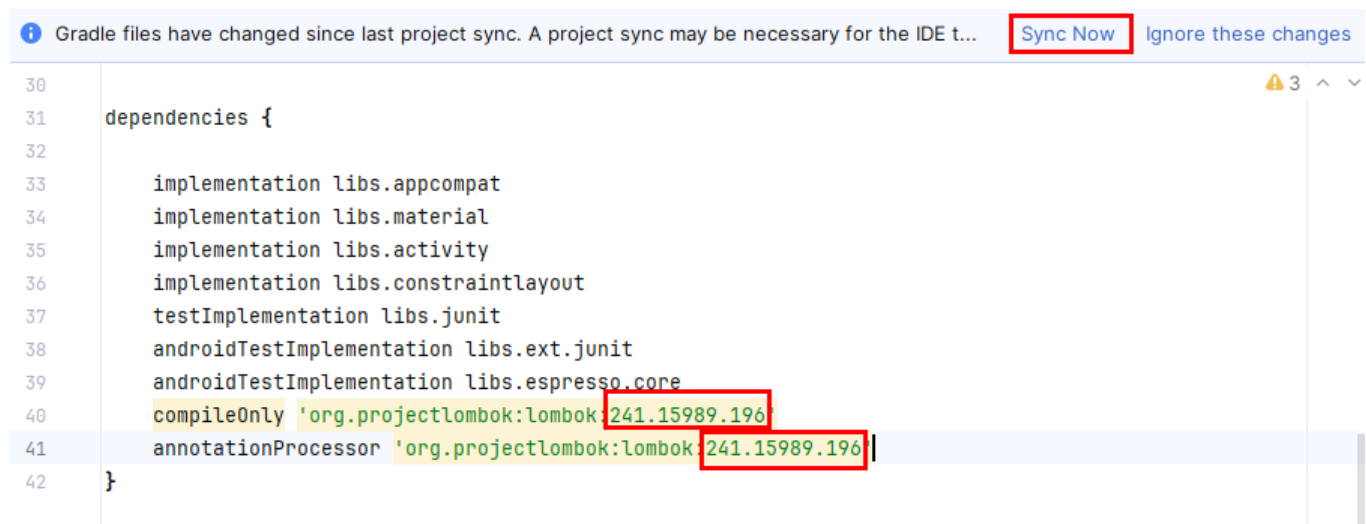
File > Settings > Plugins, tìm "Lombok" và cài đặt

Bước 2: Chỉnh dependencies

- Thêm vào file **build.gradle**

```
dependencies {  
    compileOnly 'org.projectlombok:lombok:1.18.34'  
    annotationProcessor 'org.projectlombok:lombok:1.18.34'  
}
```

- Điền đúng version đã cài đặt, và nhấn “Sync Now”

**Bước 3: Viết annotation cho Class**

Phím tắt để gọi lombok: **Alt + Insert:**

