# Java

//Classe qui hérite de «Personne» et qui implémente «CallBack»

**public class Eleve extends Personne implements** CallBack **{**

//Attribut accessible dans toute la classe élève

**private String nom;**

//Constructeur

**public Eleve() {**

}

**public Eleve(String nom) {**

**this.nom = nom;**

}

//methode

**private** ArrayList<Eleve> **get10Eleve(){**

//Déclaration d’une variable accessible uniquement dans la méthode

Eleve eleve = **null;**

//Déclaration et instanciation d’une variable

ArrayList<Eleve> eleveArrayList = new ArrayList<>();

for(int i=0; i< 10; i++) {

//Instanciation de la variable (création de l’espace mémoire).  
 e = **new Eleve();**;  
 e.setNom("Eleve" + i);

//Ajout à la fin de la liste

eleveArrayList.add(eleve);  
 }

**return** eleveArrayList;

}

//getter

**public String getNom() {**

**return nom;**

}

//setter

**public void setNom(String nom) {**

**this.nom = nom;**

}

//Implémentation des méthodes abstraites

@Override

public void onClick(final View v) {

}

}

//Classe abstraite

public abstract class Personne{

private String prenom;

//Méthode abstraite

protected abstract int getIdentifiant();

protected String getPrenom() { return prenom; }

}

//Interface

public interface CallBack{

void onClick (Eleve eleve);

}

# Activity et AsyncTask

public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
 /\* ---------------------------------  
 // Déclaration des variables  
 // -------------------------------- \*/  
 //Constantes  
 private static final String *ELEVE\_LIST\_EXTRA* = "ELEVE\_LIST\_EXTRA";  
  
 //Graphique  
 private TextView tv\_name;  
 private Button bt\_test, bt\_charger;  
  
 //Gestion d'une liste  
 private ListView lv; //le composant graphique  
 private EleveAdapter eleveAdapter;//La relation entre le composant graphique et la liste  
 private ArrayList<Eleve> eleveList;  
  
  
 /\* ---------------------------------  
 // Cycle de vie de l'activite  
 // -------------------------------- \*/  
  
 //Création de l'activité  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 //Fichier XML représentant l'activité  
 setContentView(R.layout.*activity\_main*);  
  
 //On récupère le pointeur vers le TextView  
 tv\_name = (TextView) findViewById(R.id.tv\_name);  
 bt\_test = (Button) findViewById(R.id.bt\_test);  
 bt\_charger = (Button) findViewById(R.id.bt\_charger);

lv = (ListView) findViewById(R.id.*lv*);  
  
 //création de la liste  
 eleveList = new ArrayList<>();  
 eleveAdapter = new EleveAdapter(this, eleveList);  
 //ON associe notre Listview à notre adapter  
 lv.setAdapter(eleveAdapter);  
  
 //on utilise notre composant  
 tv\_name.setText("Hello!!");  
  
 //On redirige les clics du bouton vers la classe courante  
 bt\_test.setOnClickListener(this);   
 bt\_charger.setOnClickListener(this);  
 }  
  
 //L'activité vient d'être recréée après avoir été détruite. on récupère nos valeurs

@Override  
 protected void onRestoreInstanceState(Bundle savedInstanceState) {  
 super.onRestoreInstanceState(savedInstanceState);  
 //On récupère nos objets sauvegardés  
 List<Eleve> temp = savedInstanceState.getParcelableArrayList(*ELEVE\_LIST\_EXTRA*);  
 //On efface et on remplit la liste, pour ne pas modifier le pointeur.  
 eleveList.clear();  
 eleveList.addAll(temp);  
 //On prévient notre adapter que les données ont changé.  
 eleveAdapter.notifyDataSetChanged();  
 }  
  
 //L'activité prend la main  
 @Override  
 protected void onResume() {  
 super.onResume();  
 //Ecrire dans la console  
 Log.*w*("Mon tag", "MainActivity est visible à l'écran");  
 }  
  
 //L'activité perd la main  
 @Override  
 protected void onPause() {  
 super.onPause();  
 }

//L'activité va se détruire pour être recréée. On sauvegarde nos valeurs  
 @Override  
 protected void onSaveInstanceState(Bundle outState) {  
 super.onSaveInstanceState(outState);  
 //On sauvegarde notre Liste, mais son contenant doit être Parcelable  
 outState.putParcelableArrayList(*ELEVE\_LIST\_EXTRA*, eleveList);  
 }  
  
 //L'activité est détruite  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 }  
  
  
 /\* ---------------------------------  
 // Click  
 // -------------------------------- \*/  
  
 @Override  
 public void onClick(View v) {  
  
 //2 façons d'identifier son bouton  
 if (v == bt\_test && v.getId() == R.id.bt\_test) {  
 //Lancer une autre Activité  
 Intent intent = new Intent(this, SecondActivity.class);  
 //Lui transmettre un param  
 intent.putExtra("Cle", "hello from mainActivity");  
 startActivity(intent);  
 }

else if (v == bt\_charger) {  
 //Création de l'AsyncTask  
 ChargementEleveAT chargementEleveAT = new ChargementEleveAT();  
 //On lance l'asyncTask  
 chargementEleveAT.execute();  
 }  
 }  
  
 /\* ---------------------------------  
 // AsyncTask  
 // -------------------------------- \*/  
  
 public class ChargementEleveAT extends AsyncTask {  
  
 */\*\*  
 \* Constructeur  
 \*/* public ChargementEleveAT() {  
  
 }  
  
 */\*\*  
 \* Dans un thread séparé  
 \*  
 \** ***@param*** *params Reçu de "execute()"  
 \** ***@return*** *resultat transmis à onPostExecute  
 \*/* @Override  
 protected Object doInBackground(Object[] params) {  
  
 //On lance une requête serveur pour récupèrer les élèves  
 try {  
 List<Eleve> temp = Serveur.chargerEleve();  
 return temp;  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
  
 */\*\*  
 \* Après l'execution de doInBackground et sur l'UIThread  
 \*  
 \** ***@param*** *o Resultat de doInBackground  
 \*/* @Override  
 protected void onPostExecute(Object o) {  
 super.onPostExecute(o);  
 List<Eleve> temp = (List<Eleve>) o;  
 //On modifie notre liste avec celle reçu  
 eleveList.clear();  
 eleveList.addAll(temp);  
 eleveAdapter.notifyDataSetChanged();  
 }  
 }  
}

# Lien

Rendre un objet Parcelable : <http://www.parcelabler.com/>

XML to Activity : <https://www.buzzingandroid.com/tools/android-layout-finder/>

Liste des librairies Android : <http://android-arsenal.com/free>

Balises supportées par les TextView : <http://daniel-codes.blogspot.fr/2011/04/html-in-textviews.html>

Base d’icone Material Design : <https://design.google.com/icons/>

Convertisseur DP <-> Px : <http://pixplicity.com/dp-px-converter/>

Inkscape : <https://inkscape.org/fr/>