@author akpaki

Documentation sur le projet de voiture avec deux (2) activités
---

Notre projet consiste a passer d une activite a une autre en cliquant sur un item de la premiere activite pour afficher les infos le concernant sur la deuxieme;
Sur la deuxieme activite nous avons un bouton retour "Back" qui permet de retourner sur l activite rincipale.

Ce qui nous a permis de mettre en pratique les notions d Activity, de Intent et d Objet.

-Notre projet est composé d'un package com.ingeniousafrica.android

-Dans ce package nous avons onze (11) classes dont 8 générées automatiquement et quatre (4) sources que nous avons creer. Voici les details:

> CarActivity: l activité principale (first activity) de notre projet heritant de la classe "Activity"

* Dans cette classe nous avons la methode onCreate(Bundle savedInstanceState) de type void qui est appelée quand l'activité est créée en premier lieu et permet de restaurer l etat de l interface utilisateur (IU) grace au parametre "savedInstanceState"

* Apres nous avons fixé la mise en page de l activite par son gabarit (layout) correspondant

*Recuperation de la listview creer dans le fichier xml par la methode "findViewById"

* Creation des objets de la calsse Voiture (details plus bas)

*Création de la ArrayList qui nous a permis de remplir la listView

*Creation d'une HashMap pour insérer les informations des items de notre listView

*Création d'un SimpleAdapter qui se chargera de mettre les items présents dans notre list (listItem) dans la vue(gabarit) affichageitem

* Attribution à notre listView l'adapter que l'on vient de créer avec setAdapter

* Mise en place d un écouteur d'évènement sur notre listView

*On récupère la HashMap contenant les infos de notre item (titre, description, img)

* On créé un objet Bundle, c'est ce qui va nous permetre d'envoyer des données à l'autre Activity

* Mettons les infos à transporter par couple cle/valeur grace a la methode putString

* Creation de l'Intent avec 2 parametres qui va nous permettre d'afficher l'autre Activity:
Le second prametre correspond à la cible de notre intent(InfosActivity)

*On affecte à l'Intent le Bundle que l'on a créé. La methode "putExtras" est appliqué sur un objet de type intent permet
* de transporter des infos et ayant pour principe couple

doc

* cle/valeur---> A chaque valeur correspond une clé.

* On démarre l'autre Activity grace a la methode startActivity

---

> la classe Voiture herite de l interface (implements) Serializable (une interface publique)

* Creation des variables locales marque, modele, constructeur, img et la constante private static final long serialVersionUID = 1L générée automatiquement ;

* Le constructeur Voiture() a pris 4 choses en parametres a savoir: la marque, le modele, le constructeur et l image

* Dans cette classe nous avons utilisé les methodes suivantes:

getConstructeur() qui retourne constructeur

getImg() qui retourne l image

getMarque() qui retourne la marque

getModele() qui retourne le modele

setConstructeur (java.lang.String constructeur) de type void qui attribut la variable constructeur a la variable private constructeur de type String

setImg(java.lang.String img) de type void qui attribut la variable img a la variable private img

setMarque(java.lang.String marque) de type void qui attribut la variable marque a la variable private marque

setModele(java.lang.String modele) de type void qui attribut la variable modele a la variable private modele

---

> La classe InfosActivity: notre deuxieme activité (ecran) qui servira d interaction avec l utilisateur.

* Cette classe herite de Activity et herite de l interface OnclickListener

* Toujours l activite est creée grace la methode onCreate.

* Creation d un objet de type Bundle qui nous a permis d extraire les infos de l intent envoyées par l objet Bundle de l activite principale

* On crée un écouteur d'évènement pour notre Button par setOnClickListener

* La methode onClick gere l action effectuer sur le boutton

* Creation d un objet intent afin de diriger l action du boutton vers notre activity cible (CarActivity dans notre cas)

* On démarre l'autre Activity grace a la methode

startActivity

La classe suivante est notre propre adapter creer en vue de gerer les données concernant chaque item de notre listeview

>La classe VoitureAdapter herite de la classe BaseAdapter

* On cree un objet LayoutInflater. Il a pour mission de charger notre fichier "Voitureinfos.xml" pour l'item

* Le constructeur VoitureAdapter(Context context,List<Voiture> prends 2 choses en parametres: Le contexte actuel et notre liste

les methodes utilisees dans cette classes sont:

getCount() :methode qui retourne le nombre d élements dans notre liste

getItem(int position) : methode qui retourne notre objet voiture à la position indiquée

getItemId(int position) : methode qui retourne l id de voiture

getView(int position, View convertView, ViewGroup parent) : methode qui retourne la vue de l'item pour l'affichage

. Creation d une class ViewHolder au sein de la casse VoitureAdapter qui nous a servi de mémoriser les éléments de la liste en mémoire pour qu'à chaque rafraichissement l'écran ne scintille pas

---