

TP04

Matière : ATELIER DEVELOPPEMENT MOBILE1

Classes : SEM21

"Ticket" est une application android qui permet de gérer les tickets d'un ensemble de véhicules (voiture, camion et bus) pour les transporter avec leurs passagers dans un bateau. Un Ticket est défini par un **matricule**, un **type** ("Voiture" ou "Camion" ou "Bus"), un **nombre de passager**, un **poids** (true : léger ou false : lourd) et une **remise** (en pourcentage par exemple 10%). De plus Ticket possède une méthode `getPrix()` qui retourne le prix du ticket et une méthode `getType()` qui retourne un String (Léger ou Lourd).

La description des méthodes de la classe MainActivity est donnée par le tableau suivant :

Méthode	Description
init()	<ul style="list-style-type: none"> - Initialise tous les attributs graphique (edM...lstT) - Initialise adpT et le lie à lstT - Appelle <code>effacer()</code> et <code>ajouterEcouteurs()</code>
ajouterEcouteurs()	<ul style="list-style-type: none"> - Lorsque l'utilisateur clique sur btnA appeler <code>ajouter()</code> - Lorsque l'utilisateur clique sur btnE appeler <code>effacer()</code> - Lorsque l'utilisateur clique sur btnV appeler <code>vider()</code> - Lorsque l'utilisateur clique sur un item de lstT appeler <code>afficher(position)</code>
ajouter()	<ul style="list-style-type: none"> - Si edM et edR ne sont pas vides et nbPassager>0 et remise<100 elle instancie un Ticket avec les données choisis, l'ajoute à adpT et appelle <code>effacer()</code>, sinon elle affiche le message d'erreur adéquat (toast1 ou toast2 ou toast3).
effacer()	<ul style="list-style-type: none"> - Efface edM - Sélectionne le premier élément dans spT - Met le nombre de passager à 1 - Coche rdLe (Léger) - Efface edR - Met le curseur dans edM
vider()	<ul style="list-style-type: none"> - Affiche <code>alertDialog1</code> et si l'utilisateur clique sur OUI elle vide adpT
afficher(int position)	<ul style="list-style-type: none"> - Affiche toast4 qui contient le matricule, la description, la remise et le prix du Ticket d'indice position dans lstT

Travail demandé

En utilisant les indications, Donner le code des méthodes `init()`, `ajouterEcouleurs()`, `ajouter()`, `effacer()`, `vider()` et `afficher(int position)`.

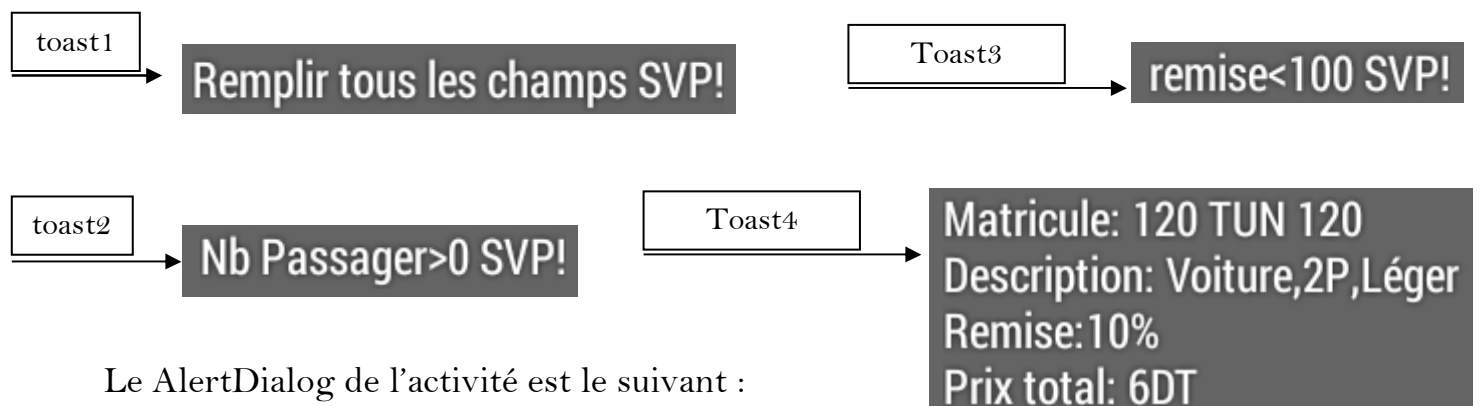
Indications

L'interface de l'activité MainActivity est la suivante :

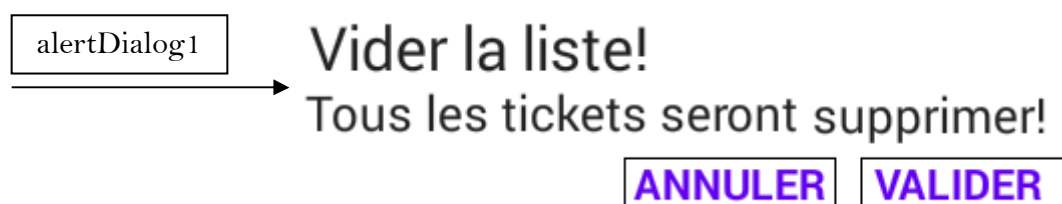
The screenshot shows the MainActivity interface with the following elements and annotations:

- Header:** A purple bar with the title "Ticket".
- Matricule:** A text field containing "120 TUN 122" with a cyan underline. An arrow labeled `edM` points to it.
- Type:** A text field containing "Voiture". An arrow labeled `spT` points to it.
- Nb Passager:** A horizontal slider with a cyan dot at the 2nd position. An arrow labeled `seekNbP` points to it.
- Poids:** Two radio buttons: "Léger" (selected with a cyan circle) and "Lourd". Arrows labeled `rdLe` and `rdLo` point to them respectively. An arrow labeled `rdgP` points to the "Lourd" button.
- Remise:** A text field containing "0 %". An arrow labeled `edR` points to it.
- Buttons:** Three purple buttons: "AJOUTER", "EFFACER", and "VIDER". An arrow labeled `btnA` points to "AJOUTER". An arrow labeled `btnE` points to "EFFACER". An arrow labeled `btnV` points to "VIDER".
- List:** A list containing two items: "120 TUN 120{Voiture- 2P- 6DT}" and "120 TUN 121{Voiture- 3P- 8DT}". An arrow labeled `lstT` points to the list.
- Annotations:** A box labeled "Contenu de spT" contains a list: "Voiture", "Camion", "Bus".

Les quatre Toasts de l'activité sont les suivants :



Le AlertDialog de l'activité est le suivant :



Le code de la classe Ticket est le suivant :

```
public class Ticket {
    private static final int[][] TARIF_VEHICULE={{5,10,15},{5,15,20}};
    private static final int TARIF_PASSAGER = 1;
    private String matricule;
    private String type;          // Voiture-Camion-Bus
    private int nbPassager;
    private boolean poidsLeger;   // true: léger   false:lourd
    private int remise;          // de 0% à 100%
    public Ticket(String matricule, String type, int nbPassager, boolean
poidsLeger, int remise) {
        this.matricule = matricule;          this.type = type;
        this.nbPassager = nbPassager;        this.poidsLeger = poidsLeger;
        this.remise = remise;
    }
    public String getMatricule() {    return matricule;    }
    public String getType() {        return type;        }
    public int getNbPassager() {      return nbPassager;  }
    public boolean isPoidLeger() {    return poidsLeger;  }
    public int getRemise() {          return remise;      }
    public int getPrix() {
        int iT=0;
        int iL=0;
        if(type.equalsIgnoreCase("Voiture"))
            iT=0;
        else if(type.equalsIgnoreCase("Camion"))
            iT=1;
        else
            iT=2;
        if(poidsLeger)
            iL=0;
        else
            iL=1;
        return (TARIF_VEHICULE[iL][iT]+TARIF_PASSAGER*nbPassager)*(100-
remise)/100;
    }
    public String getStrPoids(){
        if (poidsLeger)
            return "Léger";
        else
            return "Lourd";
    }
    @Override
    public String toString() {
        return matricule + "{" + type + "- " + nbPassager + "P"+
            "- "+ getPrix() +"DT"+ '}'';
    }
}
```

Le code de la classe MainActivity est le suivant :

```
public class MainActivity extends AppCompatActivity {
    private EditText edM;
    private Spinner spT;
    private SeekBar seekNbP;
    private RadioGroup rdgP;
    private RadioButton rdLe;
    private EditText edR;
    private Button btnA;
    private Button btnE;
    private Button btnV;
    private ListView lstT;
    private ArrayAdapter<Ticket> adpT;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
    }
    private void init() {
        ...
    }
    private void ajouterEcouteurs() {
        ...
    }
    private void ajouter() {
        ...
    }
    private void effacer() {
        ...
    }
    private void vider() {
        ...
    }
    private void afficher(int position) {
        ...
    }
}
```

Pour programmer l'évènement click sur un item d'un ListView

```
lst.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    }
});
```

Pour afficher un AlertDialog

```
AlertDialog.Builder b = new AlertDialog.Builder(this);
b.setTitle("Titre");
b.setMessage("Message");
b.setPositiveButton("Positif", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface arg0, int arg1) {
    }
});
b.setNegativeButton("Négatif", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {}
});
AlertDialog a = b.create();
a.show();
```