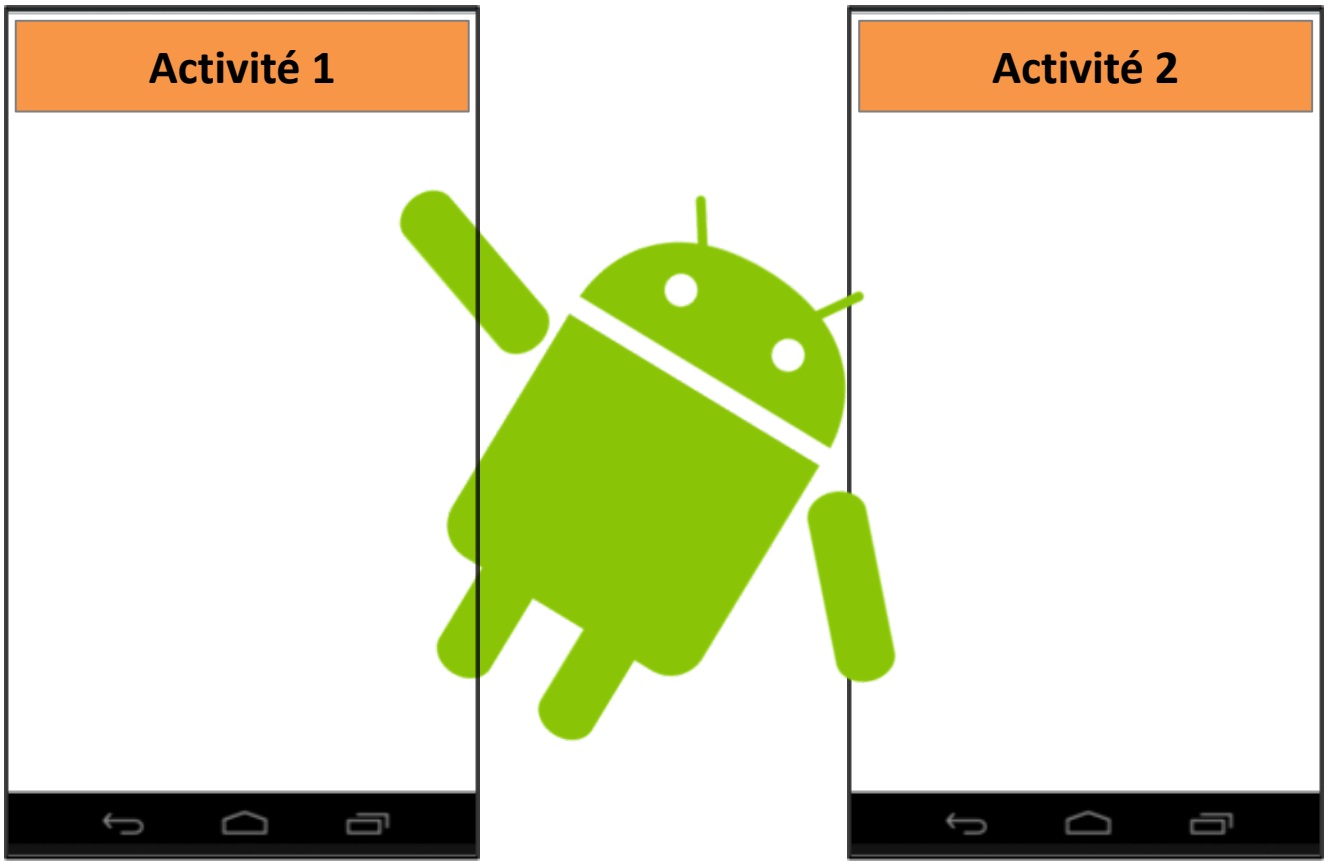
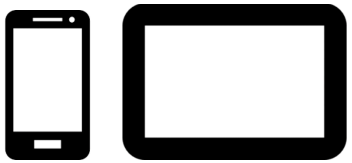
[TD2] NAVIGATION





D'UNE ACTIVITÉ À UNE AUTRE

Une application est généralement composée de plusieurs activités (plusieurs interfaces). Grâce aux **intents** nous allons pouvoir passer d'une activité à une autre.

D'UNE ACTIVITÉ À UNE AUTRE

```
public class MainActivity extends AppCompatActivity {

    protected Button b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b = findViewById(R.id.button1);
    }

    @Override
    protected void onStart() {
        super.onStart();
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent activity2;
                activity2 = new
                    Intent(MainActivity.this,MainActivity2.class);
                startActivity(activity2);
            }
        });
    }

}
```

D'UNE ACTIVITÉ À UNE AUTRE

```
public class MainActivity extends AppCompatActivity {

    protected Button b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b = findViewById(R.id.button1);
    }

    @Override
    protected void onStart() {
        super.onStart();
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent activity2;
                activity2 = new
                    Intent(MainActivity.this,MainActivity2.class);
                startActivity(activity2);
            }
        });
    }

}
```

Déclaration de l'élément graphique.

Instanciation de l'élément graphique.

Ajout d'un listener sur l'élément graphique

D'UNE ACTIVITÉ À UNE AUTRE

```
public class MainActivity extends AppCompatActivity {

    protected Button b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b = findViewById(R.id.button1);
    }

    @Override
    protected void onStart() {
        super.onStart();
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent activity2;

                activity2 = new
                    Intent(MainActivity.this,MainActivity2.class);

                startActivity(activity2);
            }
        });
    }

}
```

Déclaration du message système (i.e. Intent)

Instanciation du message en précisant le contexte et la classe de l'activité à lancer.

Lancement de la nouvelle activité.

ÉCHANGE DE DONNÉES

Le passage de données entre activités pourra s'effectuer grâce à la méthode **putExtra**.
La méthode est disponible pour les types primitifs int, string, float, double et byte.

Cette méthode possède deux paramètres :

- une clé identifiant un élément
- la valeur de l'élément

ÉCHANGE DE DONNÉES

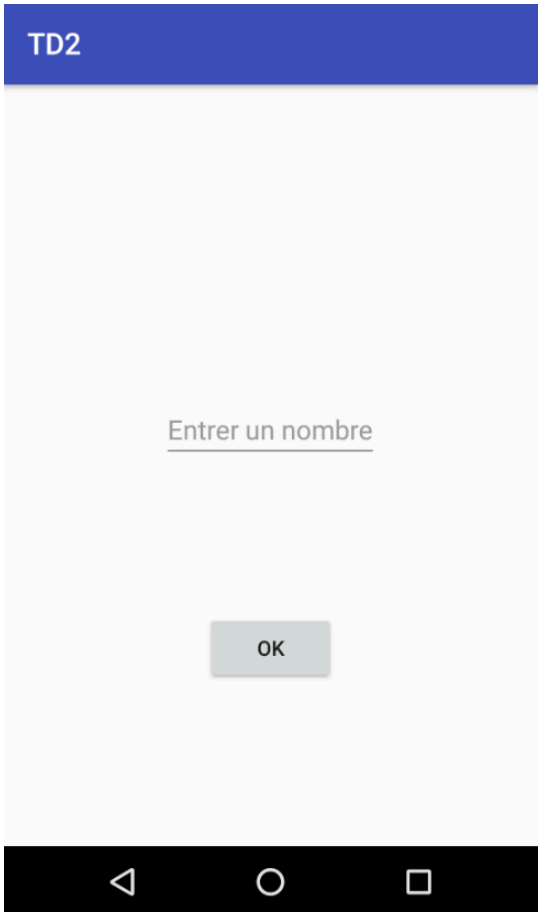
Une première activité va demander à l'utilisateur de saisir un nombre puis de le valider via un bouton.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:focusableInTouchMode="true">

    <EditText
        android:id="@+id/input"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="100dp"
        android:inputType="number"
        android:hint="@string/number" />

    <Button
        android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ok" />

</LinearLayout>
```



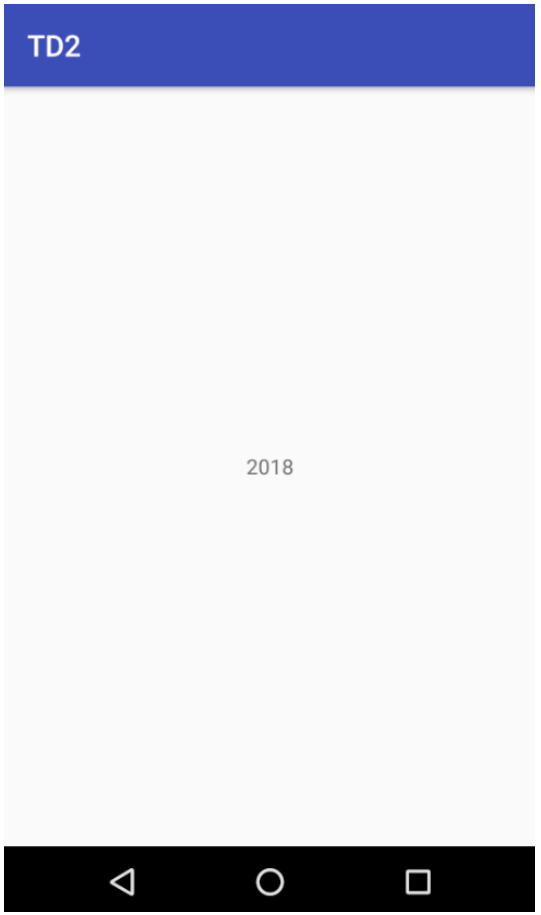
ÉCHANGE DE DONNÉES

Une deuxième activité va récupérer le nombre et va l'afficher.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:focusableInTouchMode="true">

    <TextView
        android:id="@+id/result"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"/>

</LinearLayout>
```



ÉCHANGE DE DONNÉES

```
public class MainActivity extends AppCompatActivity {

    protected Button b;
    protected EditText input;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b = findViewById(R.id.ok);
        input = findViewById(R.id.input);
    }

    @Override
    protected void onStart() {
        super.onStart();
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new
                 Intent(MainActivity.this,MainActivity2.class);
                intent.putExtra("VALUE",input.getText().toString());
                startActivity(intent);
            }
        });
    }

}
```

Dans l'activité 1, nous allons récupérer le nombre saisi par l'utilisateur et l'ajouter à l'intent grâce à la méthode putExtra.

ÉCHANGE DE DONNÉES

```
public class MainActivity2 extends AppCompatActivity {

    protected TextView message;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        message = findViewById(R.id.result);
    }

    @Override
    protected void onStart() {
        super.onStart();
        Intent intent = getIntent();
        if(intent != null) {
            message.setText(intent.getStringExtra("VALUE"));
        }
    }

}
```

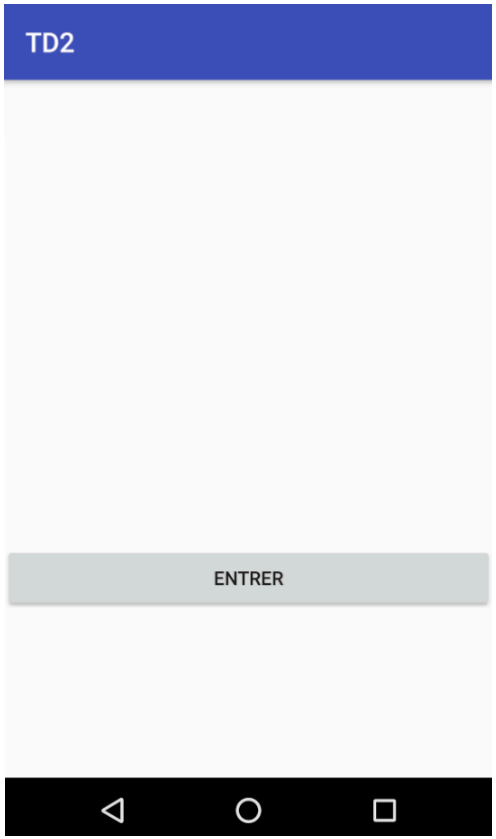
Dans l'activité 2, nous allons récupérer l'intent grâce à la méthode getIntent.

En utilisant la clef associée nous pourrons récupérer la valeur du nombre avant de l'afficher dans le TextView.

ÉCHANGE DE DONNÉES

Pour obtenir un résultat

Cette fois-ci nous déclencher une activité 2 pour lui demander que l'utilisateur saisisse un nombre.




```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">


    <TextView
        android:id="@+id/message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="100dp"
        android:gravity="center"/>

    <Button
        android:id="@+id/enter"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/launch" />

</LinearLayout>
```

ÉCHANGE DE DONNÉES

Pour obtenir un résultat

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:focusableInTouchMode="true"
    android:orientation="vertical"
    android:gravity="center">


    <EditText
        android:id="@+id/input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="100dp"
        android:gravity="center"
        android:inputType="number"
        android:hint="@string/indication"/>

    <Button
        android:id="@+id/send"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/ok" />

</LinearLayout>
```

ÉCHANGE DE DONNÉES

Pour obtenir un résultat

```
public class MainActivity extends AppCompatActivity {
    private static final int CODE = 1;
    protected TextView message;
    protected Button enter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        message = findViewById(R.id.message);
        enter = findViewById(R.id.enter);
    }

    @Override
    protected void onStart() {
        super.onStart();
        enter.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
              Intent intent =
               new Intent(MainActivity.this,Main2Activity.class);
              startActivityForResult(intent,CODE);
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode,
      int resultCode, Intent intent) {
        if((requestCode == CODE)&&(resultCode == RESULT_OK)) {
            message.setText(intent.getStringExtra("VALUE"));
        }
    }
}
```

L'activité 2 est lancée avec la méthode startActivityForResult pour préciser que l'activité 1 est dans l'attente d'un retour.

Cette portion de code sera déclenchée dès que l'activité 2 fera un retour.

En utilisant l'identifiant de l'intent nous pourrons réalisé le traitement adéquat.

ÉCHANGE DE DONNÉES

Pour obtenir un résultat

```
public class Main2Activity extends AppCompatActivity {

    protected Button ok;
    protected EditText input;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        ok = findViewById(R.id.send);
        input = findViewById(R.id.input);
    }

    @Override
    protected void onStart() {
        super.onStart();
        ok.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.putExtra("VALUE",input.getText().toString());
                setResult(RESULT_OK,intent);
                finish();
            }
        });
    }
}
```

L'activité 2 stockera dans l'intent la valeur du nombre et renverra le résultat à l'aide de la méthode setResult.

À noter que la fonction finish permet de mettre fin à l'activité 2.

Les valeurs du type de retour sont : RESULT_OK et RESULT_CANCELED

ÉCHANGE DE DONNÉES

Parcelable

L'une des limitations de transférer des données via un **Bundle** est que nous sommes restreints à des types primitifs.

Le transfert d'un objet plus complexe s'effectuera grâce un **Parcelable**.

ÉCHANGE DE DONNÉES

Parcelable

Nous allons dans cet exemple encapsuler des données dans une classe, puis faire transiter cette classe entre deux activités. L'objectif de cet exemple est donc d'illustrer le transfert de données plus complexes qu'un type primitif.

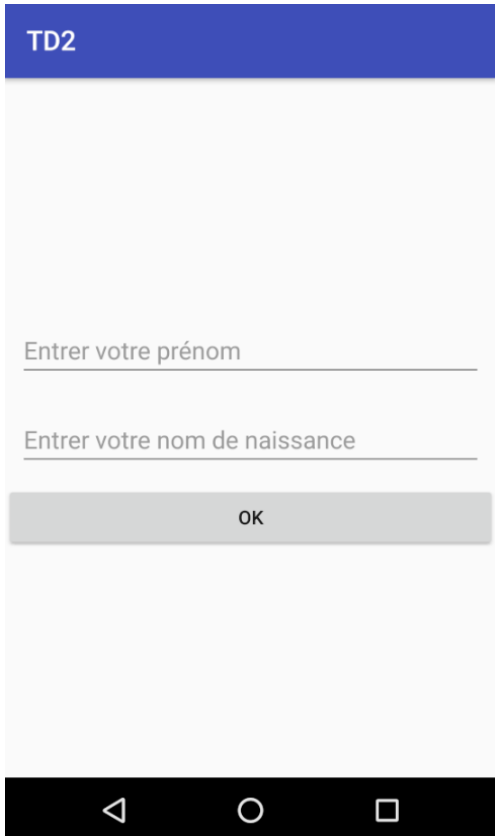
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:focusableInTouchMode="true"
    android:orientation="vertical"
    android:gravity="center">

    <EditText
        android:id="@+id/in_first_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="@string/first_name"/>

    <EditText
        android:id="@+id/in_last_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="@string/last_name"/>

    <Button
        android:id="@+id/send"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/ok" />

</LinearLayout>
```



ÉCHANGE DE DONNÉES

Parcelable

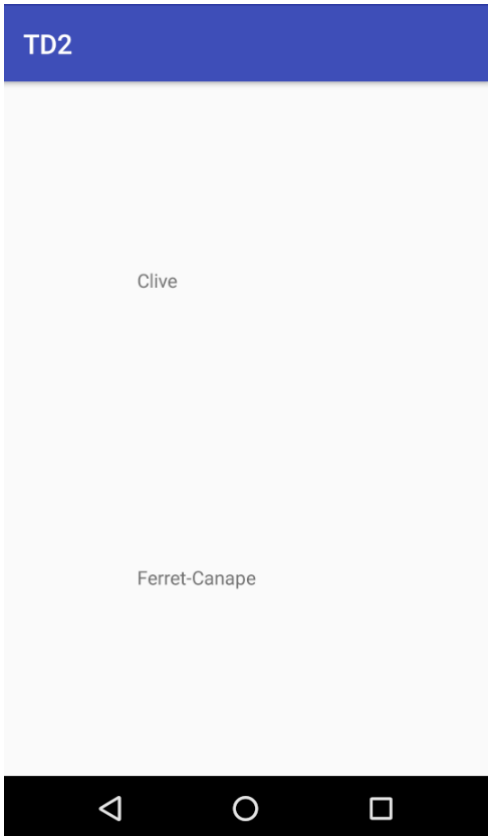
L'activité 2 qui recevra ces données sera en charge de les afficher.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:focusableInTouchMode="true"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:id="@+id/out_first_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="100dp"/>

    <TextView
        android:id="@+id/out_last_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="100dp"/>

</LinearLayout>
```



ÉCHANGE DE DONNÉES

Parcelable

```
public class B implements Parcelable {
   protected A a;
   // ...

   B(Parcel in) {
      // ...
      this.wrap = in.readParcelable(A.class.getClassLoader());
      // ...
   }
   @Override
   public void writeToParcel(Parcel dest, int flags) {
      dest.writeParcelable(this.wrap,flags);
   }

   // ...
}

public class A implements Parcelable {
   // ...
}
```

La classe est encapsulera les données héritera de la classe Parcelable.

Vous pourrez aussi faire des compostions ou des agrégations de classes.

ÉCHANGE DE DONNÉES

Parcelable

Vous devrez passer l'ensemble des données dans le constructeur de la classe héritant de Parcelable (nous son code ensuite).

L'objet sera ensuite ajouté à l'intent.

```
public class MainActivity extends AppCompatActivity {

    private EditText firstName;
    private EditText lastName;
    private Button valid;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        firstName = findViewById(R.id.in_first_name);
        lastName = findViewById(R.id.in_last_name);
        valid = findViewById(R.id.send);
    }

    @Override
    protected void onResume() {
        super.onResume();
        valid.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
             Intent intent = new Intent(MainActivity.this, Main2Activity.class);

                Person p = new
                 Person(firstName.getText().toString(),lastName.getText().toString());
                intent.putExtra("PERSON",p);
                startActivity(intent);

            }
        });
    }
}
```

ÉCHANGE DE DONNÉES

Parcelable

```
public class Main2Activity extends AppCompatActivity {

    protected TextView firstName;
    protected TextView lastName;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        firstName = findViewById(R.id.out_first_name);
        lastName = findViewById(R.id.out_last_name);
    }

    @Override
    protected void onStart() {
        super.onStart();

        Intent intent =getIntent();
        Person p = intent.getParcelableExtra("PERSON");

        firstName.setText(p.getFirstName());
        lastName.setText(p.getLastName());
    }
}
```

Vous récupérerez l'intent dans l'activité 2 puis l'instance de la classe Person grâce à sa clef.

La classe Person devra hériter de la classe Parcelable.

```
public class Person implements Parcelable {
    protected String firstName;
    protected String lastName;

    Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    Person(Parcel in) {
        this.firstName = in.readString();
        this.lastName = in.readString();
    }

    public String getFirstName() {
        return this.firstName;
    }

    public String getLastName() {
        return this.lastName;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(this.firstName);
        dest.writeString(this.lastName);
    }

    public static final Parcelable.Creator<Person> CREATOR
       = new Parcelable.Creator<Person>() {
        @Override
        public Person createFromParcel(Parcel source) {
            return new Person(source);
        }

        @Override
        public Person[] newArray(int size) {
            return new Person[size];
        }
    };
}
```

```
public class Person implements Parcelable {
    protected String firstName;
    protected String lastName;

    Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    Person(Parcel in) {
        this.firstName = in.readString();
        this.lastName = in.readString();
    }

    public String getFirstName() {
        return this.firstName;
    }

    public String getLastName() {
        return this.lastName;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(this.firstName);
        dest.writeString(this.lastName);
    }

    public static final Parcelable.Creator<Person> CREATOR
            = new Parcelable.Creator<Person>() {
        @Override
        public Person createFromParcel(Parcel source) {
            return new Person(source);
        }

        @Override
        public Person[] newArray(int size) {
            return new Person[size];
        }
    };
}
```

Vous devrez développer les attributs pour stocker les valeurs à conserver ainsi que le constructeur et les getters.

D'autres méthodes sont obligatoire : describeContents, …

ÉCHANGE DE DONNÉES

Parcelable

```
@Override
public int describeContents() {
    return 0;
}

@Override
public void writeToParcel(Parcel dest, int flags) {
    dest.writeString(this.firstName);
    dest.writeString(this.lastName);
}
```

describeContents : description du contenu du Parcel

writeToParcel : écrire le contenu du Parcel

ÉCHANGE DE DONNÉES

Parcelable

```
public static final Parcelable.Creator<Person> CREATOR
        = new Parcelable.Creator<Person>() {
    @Override
    public Person createFromParcel(Parcel source) {
        return new Person(source);
    }

    @Override
    public Person[] newArray(int size) {
        return new Person[size];
    }
};
```

Permet de créer une instance de l'objet depuis un Parcel.

ÉCHANGE DE DONNÉES

Parcelable

```
Person(Parcel in) {
        this.firstName = in.readString();
        this.lastName = in.readString();
}
```

Le constructeur lira le contenu du Parcel et assignera les valeurs à une nouvelle instance de l'objet.

ÉCHANGE DE DONNÉES

Parcelable

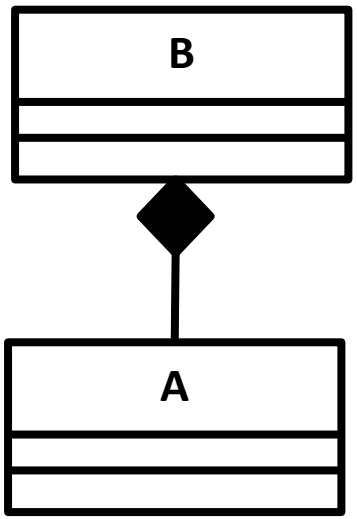
Dans le cas le où votre Parcelable serait composé lui-même d'objets, il faudra que ces objets implémentent eux aussi l'interface Parcelable.

```
public class B implements Parcelable {
    protected A a;
    // ...

   B(Parcel in) {
        // ...
        this.wrap = in.readParcelable(A.class.getClassLoader());
        // ...
    }
    @Override
    public void writeToParcel(Parcel dest, int flags) {
        // ...
        dest.writeParcelable(this.wrap,flags);
        // ...
    }

    // ...
}

public class A implements Parcelable {
    // ...
}
```



ÉCHANGE DE DONNÉES

Parcelable VS Serializable

La sérialisation, interface standard de Java *(java.io.Serializable)*, permet de rendre un objet persistant. Contrairement aux Parcelables, il ne s'agit pas d'une partie du SDK Android.

Parcelable souvent est évalué comme plus rapide que Serializable *(les benchmarks sont cependant à relativiser)*.

L'utilisation de Parcelable se justifie plus ici dans le sens où il s'agit de l'interface spécifique à Android. Nous respecterons ainsi plus la cohérence entre notre code et le SDK Android en privilégiant son utilisation.

ÉCHANGE DE DONNÉES

Parcelable VS Serializable

```
public class Person implements Parcelable {

   protected String firstName;
   protected String lastName;

   Person(String firstName, String lastName) {
      this.firstName = firstName;
      this.lastName = lastName;
   }

   Person(Parcel in) {
      this.firstName = in.readString();
      this.lastName = in.readString();
   }

   public String getFirstName() {
      return this.firstName;
   }

   public String getLastName() {
      return this.lastName;
   }

   @Override
   public int describeContents() {
      return 0;
   }

   @Override
   public void writeToParcel(Parcel dest, int flags) {
      dest.writeString(this.firstName);
      dest.writeString(this.lastName);
   }

   public static final Parcelable.Creator<Person> CREATOR
         = new Parcelable.Creator<Person>() {
      @Override
      public Person createFromParcel(Parcel source) {
         return new Person(source);
      }

      @Override
      public Person[] newArray(int size) {
         return new Person[size];
      }
   };
}
```

```
public class Person implements Serializable {

    private static final long serialVersionUID = 1L;
    protected String firstName;
    protected String lastName;

    Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return this.firstName;
    }

    public String getLastName() {
        return this.lastName;
    }

}
```

ÉCHANGE DE DONNÉES

Parcelable VS Serializable

```
public class Person implements Serializable {

    private static final long serialVersionUID = 1L;
    protected String firstName;
    protected String lastName;

    Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return this.firstName;
    }

    public String getLastName() {
        return this.lastName;
    }

}
```

Envoyer une donnée

```
Intent intent = new Intent(MainActivity.this,Result.class);
Bundle bundle = new Bundle();
bundle.putSerializable("RESULT",person);
intent.putExtras(bundle);
startActivity(intent);
```

Recevoir une donnée

```
Intent intent = this.getIntent();
Bundle bundle = intent.getExtras();
Person person = (Person) bundle.getSerializable("RESULT");
```

ÉCHANGE DE DONNÉES

Sauvegarder des données

La méthode onSaveInstanceState sera appelée avant la destruction de l'activité permettant ainsi de sauvegarder des données dans un bundle. Lorsque l'activité sera ressuscitée la méthode onRestoreInstanceState aura ce bundle en paramètre afin de recharger des données.

Si vous souhaitez récupérer un Bundle dans onResume vous pouvez procéder de la manière suivante :

```
@Override
protected void onPause() {
    super.onPause();
    getIntent().putExtra("MESSAGE","Hello world!");
}

@Override
protected void onResume() {
    super.onResume();
    Bundle bundle = getIntent().getExtras();
    if(bundle != null)
        message.setText(bundle.getString("MESSAGE"));
}
```

PM_EI5_TD2_EXO1.pdf