INTRODUCTION TO JAVA

for Android

Variables

- *int* a = 1;
- *String* b = “Android”;
- *int[]* nums = {1, 2, 3, 4};

- *ClassA* objectA = new *ClassA()*;
- *List<Apple>* apples = new *ArrayList<>()*;

- *final* int CONST = 1; *//CONST is a constant*

Loops

- int[] nums = {1, 2, 3, 4, 5};

- for (int i = 0; i < nums.length; i++){
  System.out.println(ints[i]);
  }

- *// avoid for ArrayList iterations in Android (3 times slower)*
  for (int i : nums) {
  System.out.println(i);
  }

Enums

- public enum Day {
  SUNDAY,
  MONDAY,
  TUESDAY …
  }

- Unless required, try to use **@IntDef/@StringDef** instead.

Class

```
class A {
        private int a;
        private String b = “sample String”; //field initialization

        public A(){                 // default constructor
                A(0);
        }

        public A(int a){            // designated constructor
                this.a = a;
        }

        public String sample(int a){
                // method body  …
        }
}
```

Access Specifier

- *private* – *accessible from inside the class.*
- *default* – *package level only.*
- *protected* – *package level + inherited classes.*
- public – *accessible from anywhere.*

Inheritance

- class A{
  void methodA(){ // base class method
  }
  }

- class B extends A{

  @Override
  void methodA(){ *// overridden method*
  }
  }

Abstract Class

- *abstract* class A {

```
        abstract void abstractMethod();

        void normMethod(){
                // method body …
        }
}
```

- class B extends A {

```
        void abstractMethod(){
                //method body ….
        }
}
```

Interface

- public *interface* InterfaceA{

```
        void contractMethod1();

        int contractMethod2();

        List<String> contractMethod3();
}
```

Anonymous objects

- *interface SomeListener* {
  void onListened();
  }

- *// JAVA 7 lambdas replacement …*
  new SomeListener(){

  void onListened(){
  //method code ….
  }
  }

Support Annotations

- *@Nullable/@NonNull* – nullability constraint specifier.
- *@StringRes, @ColorRes* – android resource input
- *@IntRange, @FloatRange* – param ranges
- *@IntDef/@StringDef* – annotation defined constants, *used instead of Enums*
- *@CallSuper* – forces super call

NPE management best practices …

- void customMethod(*@NonNull* String aString,
  @NonNull Object nonNullObj,
  @IntRange(min = 0, max = 10) int sample,
  @Nullable Object nullObj){

  assert aString != null; assert nonNullObj != null;
  assert sample >= 0 && sample <= 10;

  // rest of method body ….
  }

Home Work

- Create a simple Bank account simulator, such that : -
  - Customer can open an account.
  - Customer can deposit money to his/her account.
  - Customer can withdraw money from his/her account.
  - Customer can view his/her account balance.

Questions ???

Sources

- https://docs.oracle.com/javase/8/index.html
- https://developer.android.com/index.html
- https://github.com/codepath/android_guides/wiki#getting-started
- https://developer.android.com/training/articles/perf-tips.html
- https://developer.android.com/studio/write/annotations.html
- https://projectlombok.org/index.html
- https://android-arsenal.com/