

Kotlin2Java

PA Overview

Programming Assignment #3

- ▶ Using Kotlin.g4 in PA #2
 - ▶ Kotlin.g4 for parsing basic syntax of Kotlin
 - ▶ <https://kotlinlang.org/docs/reference/basic-syntax.html>
- ▶ ANTLR based translator (visitor pattern based)
 - ▶ Kotlin-to-Java (Source code-to-Source code) compiler
 - ▶ Type Inference for implicitly defined variables

Kotlin Basic Syntax

► Functions with return type inference

Kotlin

1. Function with *return*

```
fun sum(a: Int, b: Int): Int {  
    return a + b  
}
```

2. Function with an expression and inferred return type

```
fun sum(a: Int, b: Int) = a + b
```

Java

```
int sum(int a, int b) {  
    return a + b;  
}
```

Kotlin Basic Syntax

- ▶ Types of variables inferred by *rhs* expressions

Kotlin	Java
<p>val: Read-only local variables</p> <pre data-bbox="322 638 1106 842">val a: Int = 1 val b = 2 val c: Int c = 3</pre>	<p>final variables</p> <pre data-bbox="1155 638 1935 842">final int a = 1; final int b = 2; final int c; c = 3;</pre>
<p>var: Reassign-available variables</p> <pre data-bbox="322 1018 1106 1129">var x = 5 x += 1</pre>	<pre data-bbox="1155 1018 1935 1129">int x = 5; x += 1;</pre>

Kotlin Basic Syntax

- ▶ Nested functions(methods) – local classes in Java

Kotlin	Java
<pre>fun main(){ fun StringLength(obj: Any): Int? { if (obj is String) return obj.length return null } println(StringLength("String")) println(StringLength(123)) }</pre>	<pre>class Main{ public static void main(String[] args) { class Inner{ Integer StringLength(Object obj){ if (obj instanceof String) return ((String) obj).length(); return null; } } Inner inner = new Inner(); System.out.println(inner.StringLength("String")); System.out.println(inner.StringLength(123)); } }</pre>
<p><i>Result</i> 6 null</p>	

Kotlin Basic Syntax

▶ Iterating over a range

Kotlin

```
fun main(){  
    for (x in 1..5) {  
        print(x)  
    }  
    for (x in 9 downTo 0 step 3) {  
        print(x)  
    }  
}
```

Result
12345
9630

Java

```
class Main{  
    public static void main(String[] args) {  
        for(int x=1; x<=5; x++){  
            System.out.print(x);  
        }  
        for(int x=9; x>=0; x--){  
            System.out.print(x);  
        }  
    }  
}
```

Result
12345
9630

Programming Assignment #3 (Kotlin2Java)

- ▶ Develop Kotlin2Java.java with Kotlin.g4
 - ▶ Use **visitor pattern** of ANTLR for PA#3
 - ▶ Accept input and optionally output (if not specified, *output.java* is default output name) from *file-path* at command line

```
$ java Kotlin2Java input.kt [output.java]
```

- ▶ *output.java* should result in the same behavior as *input.kt*

Install Kotlin Compiler to Run APP

- ▶ Install zip/unzip

```
$ sudo apt update
```

```
$ sudo apt install zip unzip
```

- ▶ Install Kotlin

- ▶ <https://kotlinlang.org/docs/tutorials/command-line.html>

```
$ curl -s https://get.sdkman.io | bash
```

```
$ source ~/.sdkman/bin/sdkman-init.sh
```

```
$ sdk install kotlin
```

- ▶ Run Kotlin app

```
$ kotlinc hello.kt -include-runtime -d helloKT.jar
```

```
$ java -jar helloKT.jar
```

```
Hello, World!
```

```
/* hello.kt */  
fun main() {  
    println("Hello, World!")  
}
```


Reference

- ▶ Kotlin Basic Syntax

- ▶ <https://kotlinlang.org/docs/reference/basic-syntax.html>

- ▶ ANTLR

- ▶ <https://www.antlr.org/>

- ▶ The Definitive ANTLR 4 Reference – Terence Parr

Submission

- ▶ (1) Login to swin.skku.edu
- ▶ (2) Create a directory pa3
\$ mkdir pa3
- ▶ (3) Write .g4 and java files in pa3 directory
- ▶ (4) Run pl_submit command
\$ pl_submit pa3 .

- ▶ Make sure you run pl_submit command in pa3 directory.