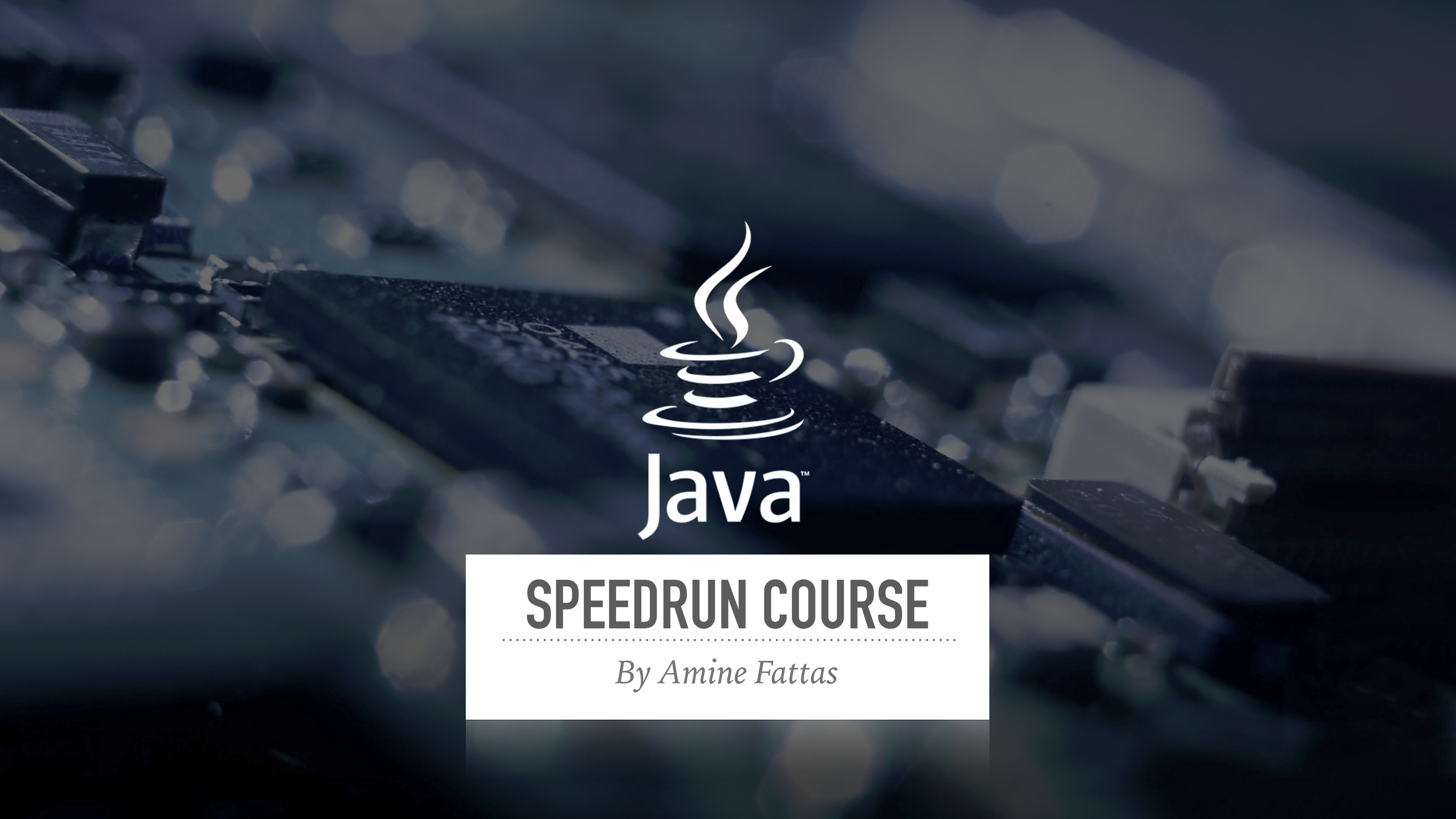
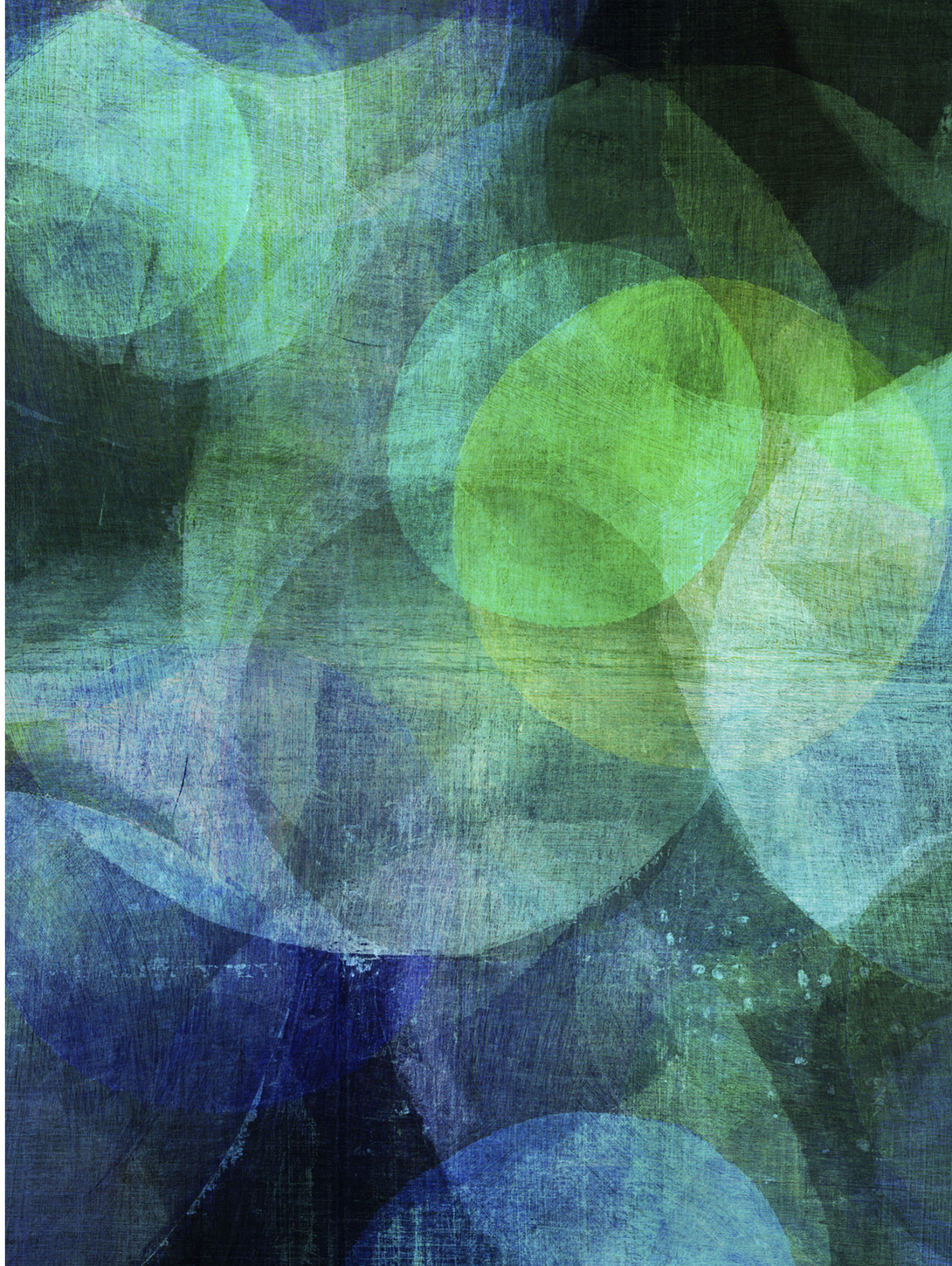


SPEEDRUN COURSE

By Amine Fattas

I – BASICS

Coding with Java



Hello World

Variables

String class

Data Structures

Flow Control

Loops

Functions

Hello World

- main method
- print in console
- import

Hello World


no usages
class HelloWorld {
no usages
public static void main(String[] args) {
System.out.println("Hello World");
// _______ END _______
System.out.println("\n");
System.out.println("_______ END _______");
}
}
Hello World
_______ END _______
Process finished with exit

Variables

- Declaration, value types, reference types, constants
- Primitives
- Casting (Transtypage)
- Basic Math
- Math Library
- Wrappers

Primitives

```
boolean bool = true;

char character = 'a';

// Numbers
byte byteVar = 1;
short shortVar = 1;
int intVar = 1;
long longVar = 1l;

float floatVar = 1.0f;
double doubleVar = 1.0;
```

Casting : byte < short < int < long < float < double

```java
// Implicit Casting
// Valid casting
short shortResult = byteVar;
float result = longVar;

// Invalid Casting
byte byteResult = shortVar;
float floatResult = doubleVar;
```

```java
// Explicit Casting
byte byteResult = (byte) shortVar;
float floatResult = (float) doubleVar;

System.out.println("result: " + result);
```

```
Casting
result: -24
```

Basic Math

```java
class MathLibrary {
    no usages
    public static void main(String[] args) {

        // Basic Operations
        System.out.println("10 + 4 = " + (10 + 4));
        System.out.println("10 - 4 = " + (10 - 4));
        System.out.println("10 * 4 = " + (10 * 4));
        System.out.println("10 / 4 = " + (10 / 4));
        System.out.println("10 % 4 = " + (10 % 4));

        System.out.println("10 / 4 = " + (10.0 / 4.0));

        int a = 10;
        a = a + 5;
        System.out.println("incremented a = " + a);
        a += 5;
        System.out.println("incremented a = " + a);

        // Incrementation
        int counter = 0;
        System.out.println("counter = " + counter++);
        System.out.println("counter = " + ++counter);
```

```
10 + 4 = 14

10 - 4 = 6

10 * 4 = 40

10 / 4 = 2

10 % 4 = 2

10 / 4 = 2.5

incremented a = 15

incremented a = 20

counter = 0

counter = 2
```

Math Lib

```
{

void main(String[] args) {

.println( "Math.abs(-4.1456)  = " + Math.abs(-4.1456)  );
.println( "Math.ceil(4.1456)  = " + Math.ceil(4.1456)  );
.println( "Math.floor(4.1456) = " + Math.floor(4.1456) );
.println( "Math.round(4.1456) = " + Math.round(4.1456) );
.println( "Math.max(5, 10)    = " + Math.max(5, 10)    );
.println( "Math.min(5, 10)    = " + Math.min(5, 10)    );
.println( "Math.exp(1)        = " + Math.exp(1)        );
.println( "Math.pow(10, 4)    = " + Math.pow(10, 4)    );
.println( "Math.sqrt(9)       = " + Math.sqrt(9)       );
.println( "Math.PI = π        = " + Math.PI            );
.println( "Math.sin(π/2)      = " + Math.sin(Math.PI/2));
.println( "Math.sin(π/2)      = " + Math.cos(Math.PI/2));
.println( "Math.tan(π/2)      = " + Math.tan(Math.PI/2));

.println( "Random float in [0,1] = " + Math.random());
.println( "Random float in [0,20] = " + Math.random() * 20);
.println( "Random int   in [0,20] = " + (int) (Math.random() * 20));

_ END ________
.println("\n");
.println("________ END ________");
```

```
Math.abs(-4.1456)  = 4.1456

Math.ceil(4.1456)  = 5.0

Math.floor(4.1456) = 4.0

Math.round(4.1456) = 4

Math.max(5, 10)    = 10

Math.min(5, 10)    = 5

Math.exp(1)        = 2.7182818284590455

Math.pow(10, 4)    = 10000.0

Math.sqrt(9)       = 3.0

Math.PI = π        = 3.141592653589793

Math.sin(π/2)      = 1.0

Math.sin(π/2)      = 6.123233995736766E-17

Math.tan(π/2)      = 1.633123935319537E16

Random float in [0,1] = 0.4022363050729939

Random float in [0,20] = 12.571126723239722

Random int   in [0,20] = 11
```

Wrappers

```
int primitiveInt = 1; // value type
Integer objectInt = 1;    // reference type

primitiveInt = null; // Primitives cannot be null
objectInt = null;

// Other Wrappers
// Boolean, Character, Byte, Short, Long, Float, Double ...
```

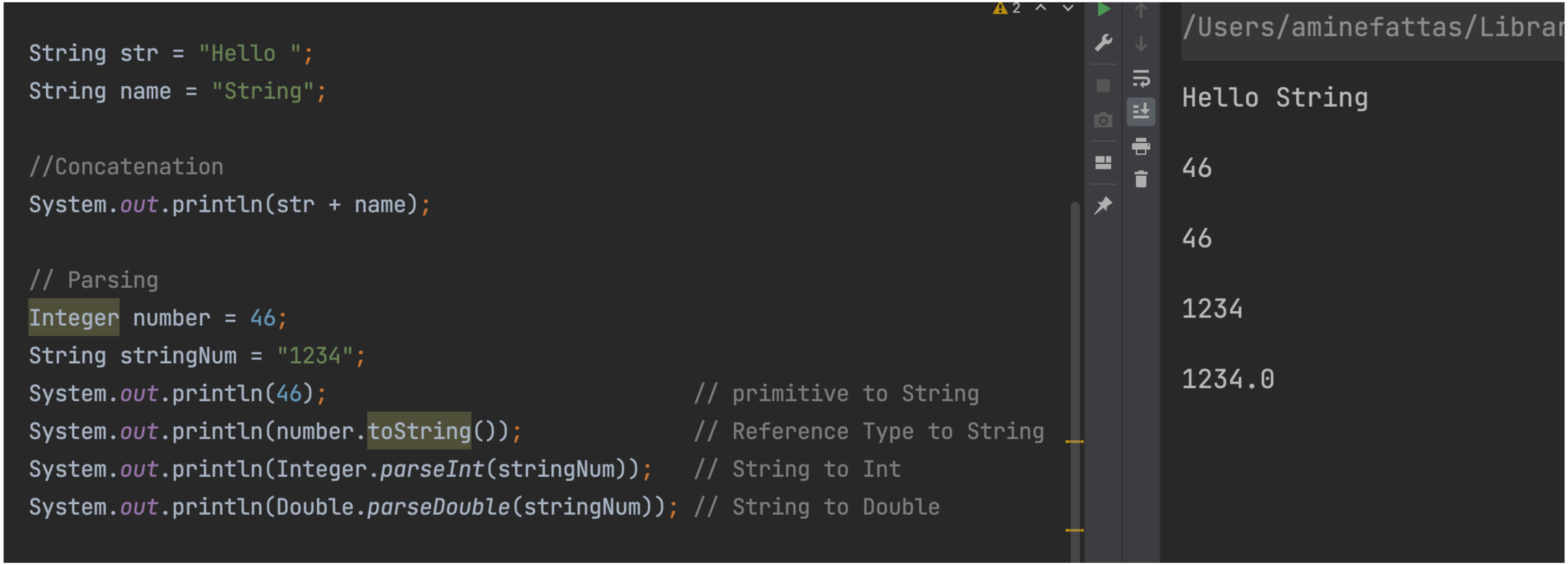
“

The scope of a local variable is the block It was declared in

String class

- Declaration
- Concatenation
- Parsing
- important methods
- StringBuilder & StringBuffer

String Concatenation & Parsing



```java
String str = "Hello ";
String name = "String";

//Concatenation
System.out.println(str + name);

// Parsing
Integer number = 46;
String stringNum = "1234";
System.out.println(46);                          // primitive to String
System.out.println(number.toString());           // Reference Type to String
System.out.println(Integer.parseInt(stringNum));   // String to Int
System.out.println(Double.parseDouble(stringNum)); // String to Double
```

```
/Users/aminefattas/Libra
Hello String
46
46
1234
1234.0
```

Important methods

```
String str = "  Hello World  ";
String str2 = new String( original: "  Hello World  ");

System.out.println("charAt 7: "   + str.charAt(7));
System.out.println("indexOf W: "  + str.indexOf('W'));
System.out.println("contains: " + str.contains("Hello"));
System.out.println("length: "   + str.length());

System.out.println("== : " + (str == str2));
System.out.println("equals: " + str.equals(str2));

System.out.println("replace: " + str.replace(
        target: "World",
        replacement: "Universe"
));

System.out.println("substring: " + str.substring(5,10));

for(String word: str.split( regex: " "))
    System.out.println("split: " + word + ",");

System.out.println("toUpperCase: " + str.toUpperCase());
System.out.println("trim: " + str.trim());
```

```
charAt 7:

indexOf W: 8

contains: true

length: 15

== : false

equals: true

replace:   Hello Universe

substring: lo Wo

split: ,

split: ,

split: Hello,

split: World,

toUpperCase:   HELLO WORLD

trim: Hello World
```

StringBuilder & StringBuffer

```java
String str = "Hello ";

StringBuilder stringBuilder = new StringBuilder(str);

stringBuilder.append("World1 ");
stringBuilder.append("World2 ");
stringBuilder.append("World3 ");
stringBuilder.append("World4 ");
System.out.println(stringBuilder);

stringBuilder.insert( offset: 0,  str: "Big ");
System.out.println(stringBuilder);

stringBuilder.delete(0,2);
System.out.println(stringBuilder);
```

```
/Users/aminefattas/Library/Java/JavaVi

Hello World1 World2 World3 World4

Big Hello World1 World2 World3 World4

g Hello World1 World2 World3 World4



_______ END _______


Process finished with exit code 0
```

Data Structures

- Arrays
- ArrayLists
- Stacks
- Hashmaps

Arrays

```java
int[] numbers1 = {1, 15, 42, 7, 4, 5};
System.out.println(numbers1[3]);

int numbers2[] = {1, 15, 42, 7, 4, 5};
System.out.println(numbers1[2]);

int[] numbers3 = new int[6];
numbers3[5] = 10;
System.out.println(numbers3[5]);
System.out.println(numbers3[4]);

System.out.println(numbers3.length);

int[][] matrix = {
        { 43, 54, 234 },
        { 656, 34, 98 }
};
System.out.println(matrix[1][2]);

int copyOfNumbers[] = Arrays.copyOf(numbers1, newLength: 6);
System.out.println(copyOfNumbers.equals(numbers1));

Arrays.sort(numbers1);
System.out.println(Arrays.toString(numbers1));
```

```
/Users/aminefattas/Library/Java/Ja
7
42
10
0
6
98
false
[1, 4, 5, 7, 15, 42]


_______ END _______
```

ArrayList

```java
ArrayList<String> arrayList = new ArrayList<String>();

ArrayList<String> arrayListWithCapacity = new ArrayList<String>( initialCapacity: 10);

ArrayList<String> arrayListWithInitValues =
        new ArrayList<String>(Arrays.asList("Monday", "Tuesday", "Wednesday"));

arrayList.add("My name");
arrayListWithInitValues.add("Thursday");
arrayListWithInitValues.remove( o: "Tuesday");

arrayListWithInitValues.set(0, "Sunday");
System.out.println(arrayListWithInitValues.toString());
System.out.println(arrayListWithInitValues.get(2));

// ________ END ________
System.out.println("\n");
System.out.println("________ END ________");
```

```
[Sunday, Wednesday, Thursday]

Thursday



________ END ________


Process finished with exit code 0
```

Stacks

```java
Stack<String> stack = new Stack<>();

stack.add("Monday");
stack.add("Tuesday");
stack.add("Wednesday");
stack.add("Thursday");

System.out.println(stack.toString());

System.out.println(stack.pop());
System.out.println(stack.toString());
```

```
/Users/aminefattas/Library/Java/JavaVi
[Monday, Tuesday, Wednesday, Thursday]
Thursday
[Monday, Tuesday, Wednesday]


_______ END _______
```

HashMap

```java
HashMap<Integer, String> months = new HashMap<>();

months.put(1, "January");
months.put(2, "February");
months.put(3, "March");
months.put(4, "April");
months.put(5, "May");

System.out.println(months.get(3));

HashMap<String, String> days = new HashMap<>();

days.put("Mo", "Monday");
days.put("Tu", "Tuesday");
days.put("We", "Wednesday");
days.put("Th", "Thursday");

System.out.println(days.get("Tu"));
```

```
/Users/aminefattas/Library/Java/JavaVi

March

Tuesday



_______ END _______


Process finished with exit code 0
```

Flow Control

- if else
- Ternary operator
- switch, case, break, default
- Exception handling, try catch finally

Flow Control:

```java
class FlowControl {
    public static void main(String[] args) {
        // if else
        int age = 25;

        if (age > 18) {
            System.out.println("Adult");
        }
        else {
            System.out.println("Minor");
        }

        // Ternary Operator
        String message = (age > 18) ? "Adult" : "Minor";
        System.out.println(message);

        // switch case
        String today = "Th";

        switch (today) {
            case "Mo":
                System.out.println("It's Monday");
                break;
            case "Tu":
                System.out.println("It's Tuesday");
                break;
            case "We":
                System.out.println("It's Wednesday");
                break;
            case "Th":
                System.out.println("It's Thursday");
                break;
            default:
                System.out.println("It's Weekend");
                break;
        }
```

```
Adult

Adult

It's Thursday



_______ END _______


Process finished with exit code 0
```

Exception Handling: try, catch, finally

```java
try {
    System.out.println(  10/0 );
}
catch (Exception exception){
    System.out.println(  exception.toString() );
}

finally {
    System.out.println( "Calcul finished" );
}
```

```
/Users/aminefattas/Library/Java/JavaVirtualMa

java.lang.ArithmeticException: / by zero

Calcul finished



_______ END _______


Process finished with exit code 0
```

> In the catch block argument, It is not recommended to define your exception as vague Exception type

```
catch (ArithmeticException exception){ // Type definition Recommended
    System.out.println(  exception.toString() );
}
catch (Exception exception){ // Not recommended
    System.out.println(  exception.toString() );
}
```

Exception Handling: throw

```
try {
    int denominator = 0;
    System.out.println(  10/denominator );
    if (denominator == 0)
        // Manually throw an exception
        throw new ArithmeticException("Cannot divide by )");
}
catch (ArithmeticException exception){ // Type definition Recommended
    System.out.println(  exception.toString() );
}
```

Loops

- for
- Enhanced for
- while
- do while

For loop

```java
ArrayList<Integer> list =
        new ArrayList<>(Arrays.asList(123, 5, 42, 678, 54));

// Normal For Loop
System.out.println("Normal For Loop");

for (int i = 0; i < list.size() - 1; i++) {
    System.out.println(list.get(i));
}

// Enhanced For Loop
System.out.println("Enhanced For Loop");

for (int num : list) {
    System.out.println(num);
}
```

```
Normal For Loop
123
5
42
678
Enhanced For Loop
123
5
42
678
54
```

While loops

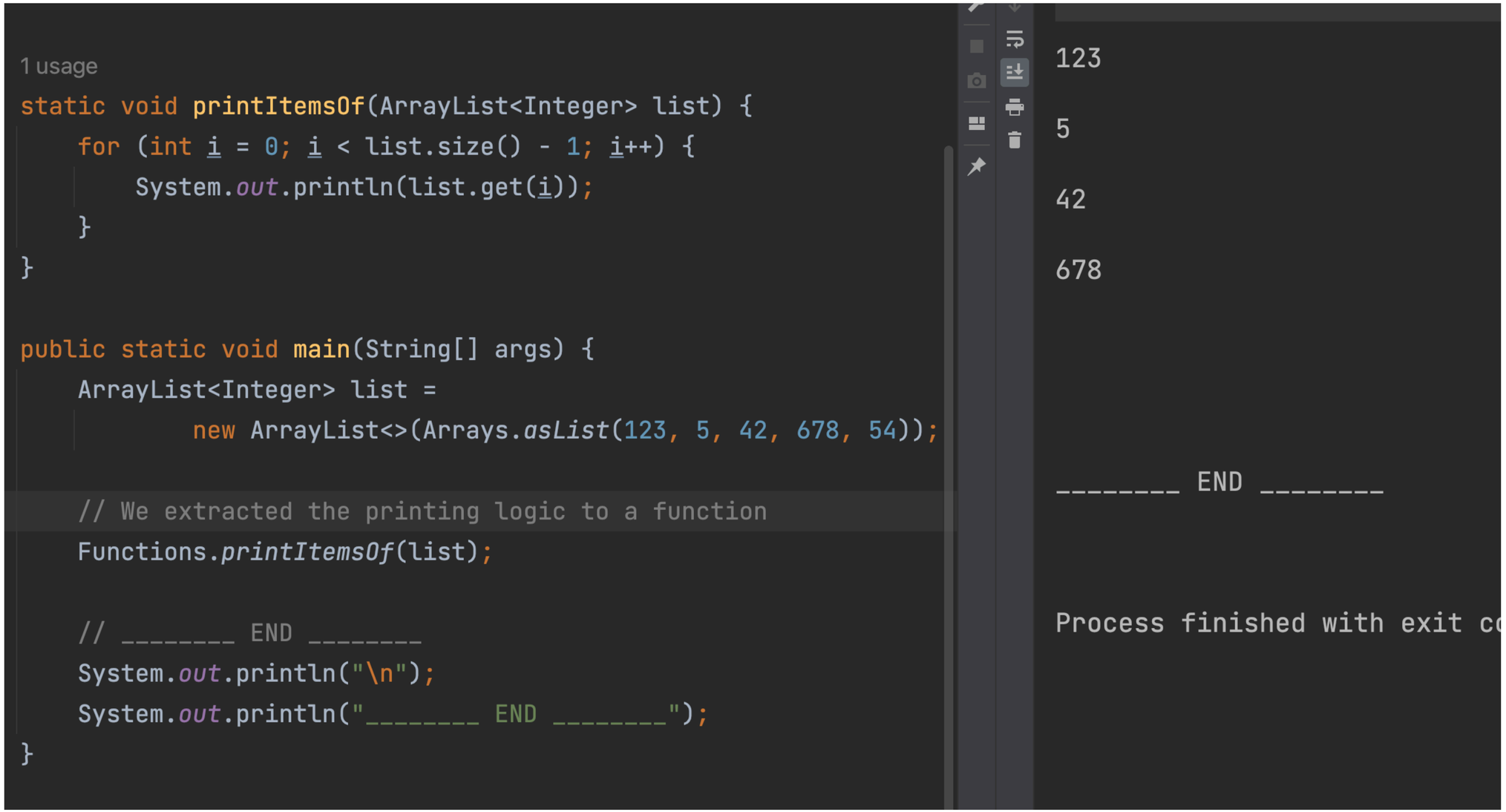
```java
class WhileLoops {
    public static void main(String[] args) {

        ArrayList<Integer> list =
                new ArrayList<>(Arrays.asList(123, 5, 42, 678, 54));

        // While
        System.out.println("While");

        int i = 0;
        while (i<list.size()) {
            System.out.println(list.get(i));
            i++;
        }

        // Do While
        System.out.println("Do While");

        int j = 0;
        do  {
            System.out.println(list.get(j));
            j++;
        } while (j<list.size());

        //          END
```

```
While
123
5
42
678
54
Do While
123
5
42
678
54
```

Functions

- Syntaxe
- Multiple Arguments (args ...)
- Recursive functions
- Throwing Functions

Syntaxe: returnType + functionName + Arguments + CodeBlock

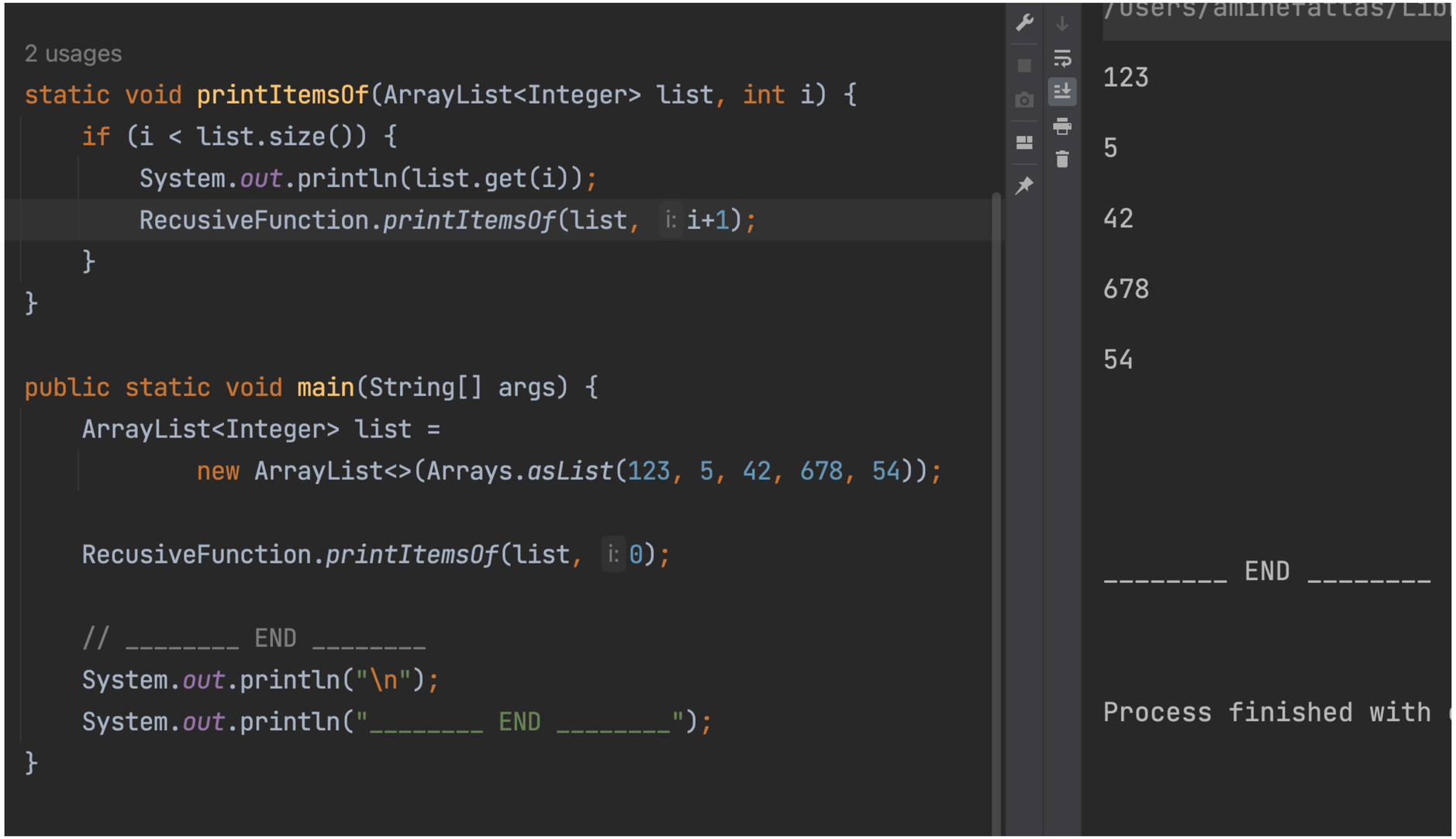


```
1 usage
static void printItemsOf(ArrayList<Integer> list) {
    for (int i = 0; i < list.size() - 1; i++) {
        System.out.println(list.get(i));
    }
}

public static void main(String[] args) {
    ArrayList<Integer> list =
            new ArrayList<>(Arrays.asList(123, 5, 42, 678, 54));

    // We extracted the printing logic to a function
    Functions.printItemsOf(list);

    // ________ END ________
    System.out.println("\n");
    System.out.println("________ END ________");
}
```

```
123

5

42

678



________ END ________


Process finished with exit co
```

Multiple Arguments

Recursive functions



```
2 usages
static void printItemsOf(ArrayList<Integer> list, int i) {
    if (i < list.size()) {
        System.out.println(list.get(i));
        RecusiveFunction.printItemsOf(list, i: i+1);
    }
}

public static void main(String[] args) {
    ArrayList<Integer> list =
            new ArrayList<>(Arrays.asList(123, 5, 42, 678, 54));

    RecusiveFunction.printItemsOf(list, i: 0);

    // ________ END ________
    System.out.println("\n");
    System.out.println("________ END ________");
}
```

```
/Users/aminefattas/Lib
123
5
42
678
54


________ END ________


Process finished with
```

Throwing functions: throws

```
static void divideByZero() throws ArithmeticException {
    System.out.println(  10/0 );
}


public static void main(String[] args) {
    try {
        ThrowingFunctions.divideByZero();
    }
    catch (ArithmeticException exception){ // Type definition Recommended
        System.out.println(  exception.toString() );
    }
```

LET'S PRACTICE !

Exercise 1

Moroccan national football team stops at Mamounia Hostel rooms:

101	102	104	202	205	211	303	304	310	404
Hakimi	Saiss	Ziyech	Ounahi	Nesyri	Bono	Yamiq	Boufal	Regragui	Amrabat

A. Represent the data in an appropriate collection.

B. Print All the players sorted by their room numbers.

C. Get the player name is in room 202.

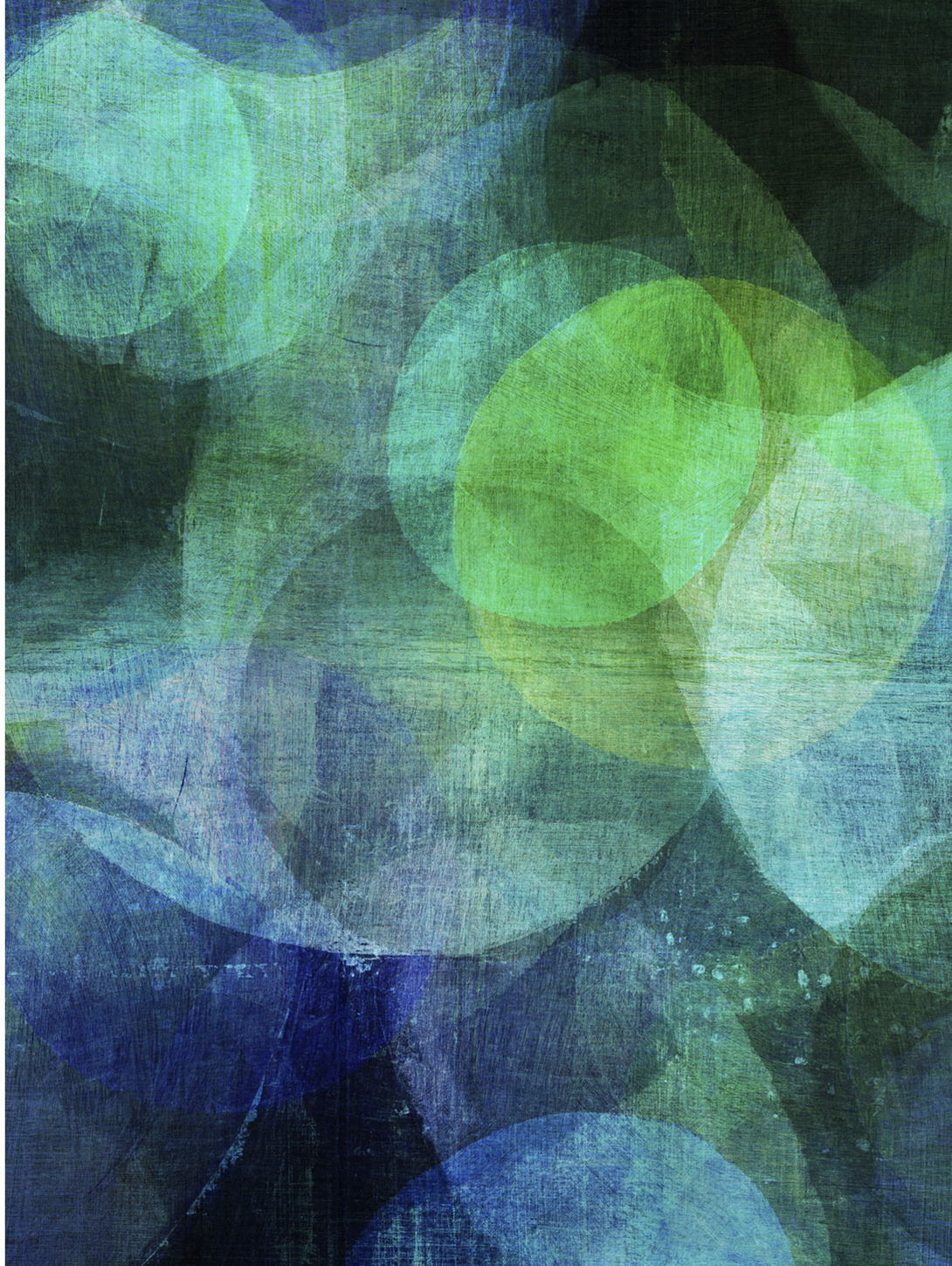
D. "Regragui" is not a player ! Replace it by "Sabiri"F

E. There was an input error ! "Amrabat" is actually in room 405. Fix it.

F. Get the player(s) whose name(s) start with 'B'.

II – INTRO TO OOP

Object Oriented Programming



Classes

Encapsulation

Inheritance

Polymorphism

Abstraction

Interfaces

Generics

Classes

- Properties
- Methods
- Constructor
- Garbage Collector
- toString()
- Instance properties/methods vs Class properties/methods
- Naming Conventions

Naming Conventions

Constant	SCREAMING_SNAKE_CASE
Variable Method	camelCase
Class Interface Enum	PascalCase

Exercise 2

Moroccan national football team stops at Mamounia Hostel rooms:

101	102	104	202	205	211	303	304	310	404
Hakimi	Saiss	Ziyech	Ounahi	Nesyri	Bono	Yamiq	Boufal	Regragui	Amrabat

A. Represent the data as **Objects**.

B. Print All the players sorted by their room numbers.

C. Get the player name is in room 202.

D. "Regragui" is not a player ! Replace it by "Sabiri"F

E. There was an input error ! "Amrabat" is actually in room 405. Fix it.

F. Get the player(s) whose name(s) start with 'B'.

Encapsulation

- Access Control
- Getters & Setters

“

A public class must have the same name of its file

Inheritance

- Syntaxe and application

Polymorphism

- Method Overriding (Redéfinition)
- Method Overloading (Surcharge)

Abstraction

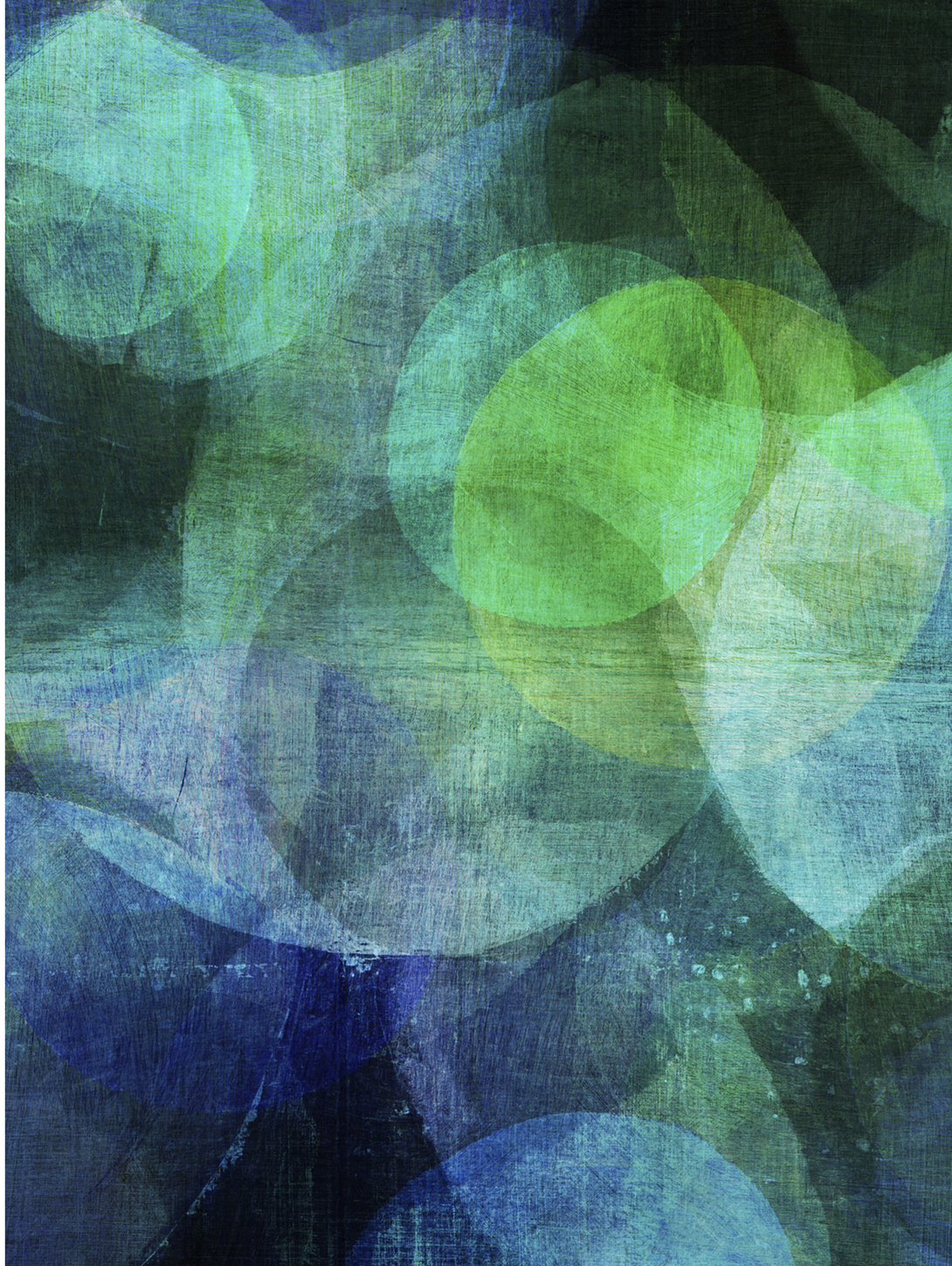
- Abstraction example with polymorphism
- abstract Class

Interfaces

- As a Contract
- Abstraction
- Delegation

III – NEXT STEPS

It's Google time !



Some common topics often found in Real Projects:

Lambda Expressions

URL Format

Dates

Regex

Design Patterns (Singleton, MVC, ...)

Maven

Threads

THANK YOU
BEST OF LUCK