Workshop on: Java Fundamentals

Rahul Dhangar • Day 2

Recap: Day 1

- Writing Our First Java Program
- Decision Structures
- Loops
- Methods

Day 2 - Topics

- Classes and Objects
- Arrays
- Text processing

Let’s start!

Classes and Objects

Objects:

Objects are structures which contain data and behaviors.

P1: Rectangle

Write a class that describes the fields and methods of a rectangle.

Encapsulation

All data and behavior is contained within the object itself.

Expose behavior and restrict direct access to data.

Constructors

Structures within a class that are used to set the initial state of an object.

Instantiating Objects

Create an instance of the Rectangle class to represent an object of that type

P2: Area of Rooms

Write a class that creates instances of Rectangle to find the total area of two rooms in a house.

Sending & Receiving Objects

Objects can be passed into methods and returned from methods

P3: Area of Rooms

Redo HomeAreaCalculator to include a calculateArea() method that accepts two Rectangles and a getRoom() method that returns a Rectangle.

Records

A special type of Java class that is great for simple objects

Creating a Record

```
1 public record Account(
2     int id,
3     int customerId,
4     String type,
5     double balance) {}
```

Adding Methods to Records

```
1  public record Account(
2    int id,
3    int customerId,
4    String type,
5    double balance) {
6
7    public void addedMethod() {
8      //...
9    }
10
11 }
```

Working with Records

```
1 Account account = new Account(12345, 54321, "CHECKING",
  2178.5);
2
3 account.setType("SAVINGS"); //gives compilation error
4
5 double balance = account.balance();
```

Wrapper Classes

```
int number1 = 5;

Integer number2 = 5;
```

Wrapper Classes

Primitive Data Type	Wrapper Class
int	Integer
double	Double
long	Long
float	Float
boolean	Boolean
char	Char
byte	Byte

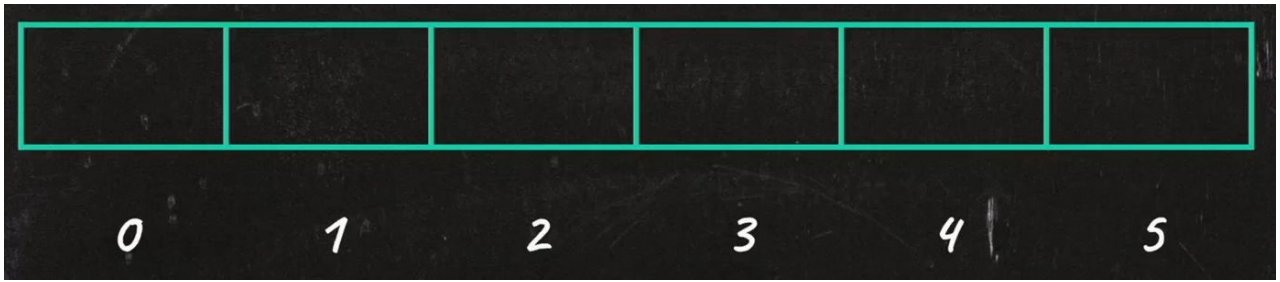
Arrays

Special variables that can hold multiple values

```
int[] lottoTicket = new int[6];
```

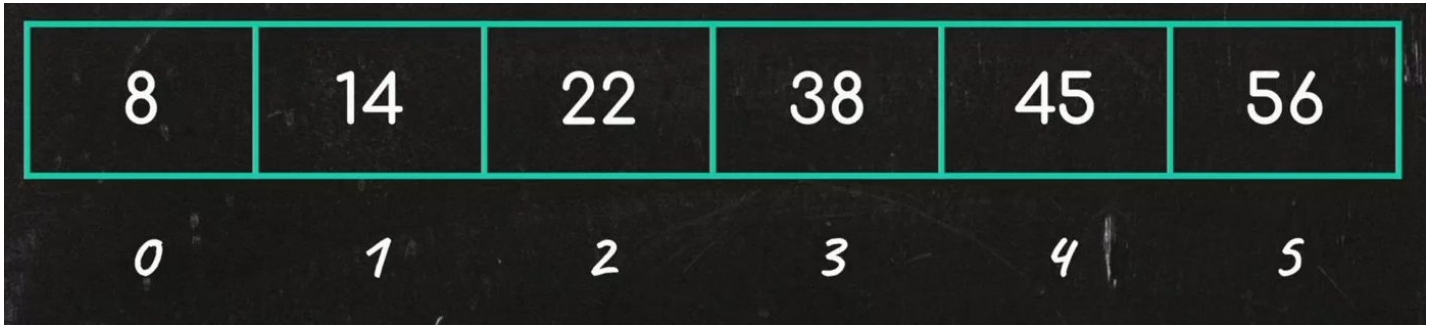
Array Elements

```
int[] lottoTicket = new int[6];
```



Array Elements

```
int[] lottoTicket = {8, 14, 22, 38, 45, 56};
```



P4: Lottery Ticket

Create a Lottery Quick Pick application that will generate a lottery ticket with 6 random numbers between 1-69.

Text processing

Strings

A sequence of characters

P5: Word Count

Write a method that counts the number of words in a String and prints them individually on new lines.

P6: Backwards String

Write a method that prints a given String backwards. For example, if given “pots”, it prints “stop”

String Builder

Used to create a mutable String that can be modified

P7: Jumbled String

Write a method that adds spaces to a jumbles String where all words were written together with no spaces. Each new word begins with a capital letter.

Text blocks

Used to work with multi-line Strings

```
String response =
"[\n" +
"  {\n" +
    "    \"id\": 13344,\n" +
    "    \"customerId\": 12212,\n" +
    "    \"type\": \"CHECKING\",\n" +
    "    \"balance\": 4022.93\n" +
    "  }\n" +
    "]";
```

Before

Text blocks

Text blocks

Used to work with multi-line Strings

```
1  return """
2          [
3            {
4              "id": 13344,
5              "customerId": 12212,
6              "type": "CHECKING",
7              "balance": 3821.93
8            }
9          ]
10       """;
```

After

Text blocks

Text blocks

```
return """ Hey y'all! """;
```

```
return """
       Hey y'all! """;
```

```
return """
       Hey y'all!
       """;
```

Questions ?