WELCOME BACK
&
THANK YOU

Advance Java
Crash Course

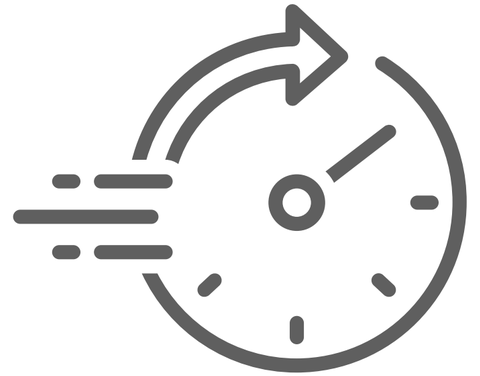
For TD Bank

MEET YOUR CRASH COURSE TEAM

Pre Assessment 10-15 min

Rapid Review

TRIVIA



Rapid Trivia

True or False, during the 1990s, did a rocket crashed in the Atlantic ocean due to software bug.

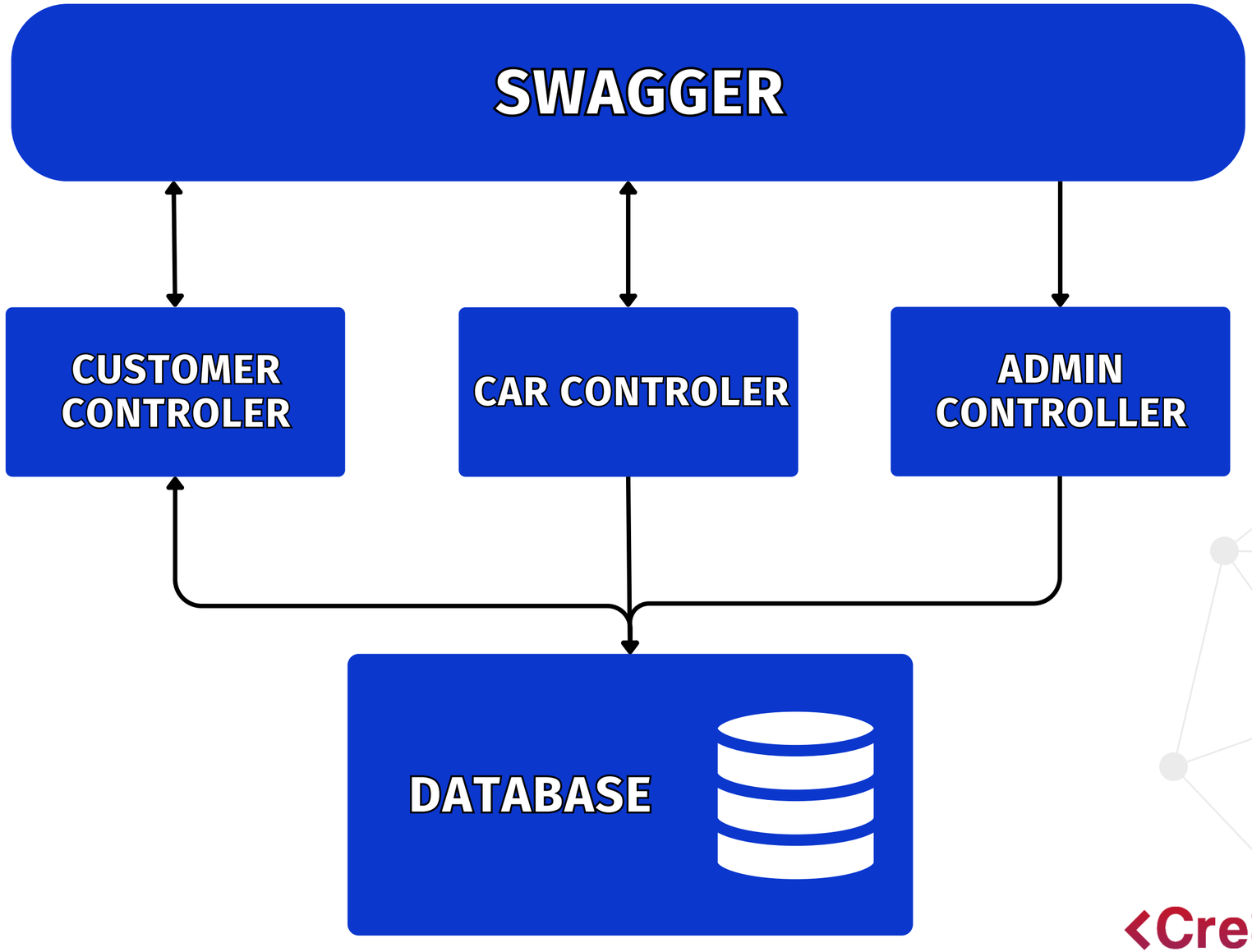
TRAINING DAY 9

TODAY'S AGENDA

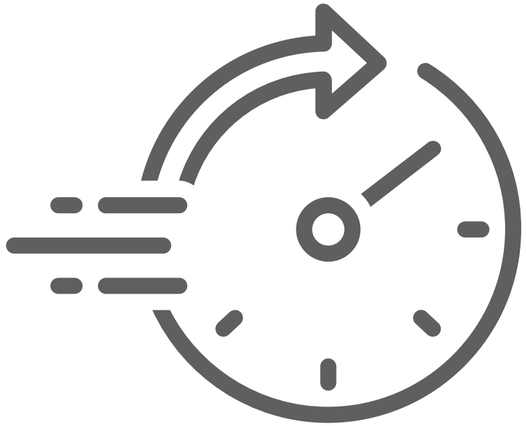
1. Yesterday's Coding Exercise to Go Rapid Review
2. Troubleshooting With Your IDE
3. Navigating Code with your IDE
4. Java Reflection
5. Java Introspection

BITE SIZE
CAR RENTAL APP

WE ARE BUILDING



Rapid Review
Lesson 1



Rapid Review

Java Reflection class name via configuration

YERSTERDAYS CODING EXERCISE TO GO

In our **com.mains.Reflection.ReflectionMain** class we instantiate a class defined in **application.properties class.className**

Create another class like **TheClass** but should return different values in their methods. Change **application.properties** to reflect the new class and execute **com.mains.Reflection.ReflectionMain**. Confirm the new values for **Id**, **Name** and **Number** should print in the terminal output.

application.properties

```
class.className=com.mains.Reflection.TheClass
```

YESTERDAY'S CODING TO GO REVIEW

```java
public final class TheOtherClass implements ClassInterface {
    private int number;
    private String name;

    public TheOtherClass() {

    }
    public int getId() {
        return 2;
    }
    @Override
    public int getNumber() {
        number = 22;
        return number;
    }
    @Override
    public String getName() {
        name = "Charlie";
        return name;
    }

}
```

```
class.className=com.mains.Reflection.TheClass
```

```
Id = 1 Name = John Number = 10
```

```
class.className=com.mains.Reflection.TheOtherClass
```

```
Id = 2 Name = Charlie Number = 22
```

DAY 8
LESSON 2

Troubleshooting Code With Your IDE

TROUBLESHOOTING WITH YOUR IDE

Breakpoints

```
28    public ResponseEntity<List<CarResponseDto>> getAllCars() {
29        List<CarResponseDto> cars = carService.getAllCars(); carService = CarService@104
30        return new ResponseEntity<>(cars, HttpStatus.OK);
```

Conditional Breakpoints

```
29        List<CarResponseDto> cars = carService.getAllCars();
```

Expression: `carService.getAllCars() == null;`

Inspecting Variables and changing variables values

```
WATCH
> carId: Long@155 "3"
```

```
WATCH
> carId = (long) 2: Long@170 "2"
```

Call Stack

```
Thread [http-nio-...  PAUSED ON BREAKPOINT
    CarService.getCarById(Long)        CarService.java
    CarController.getCarById(Long)     CarController...
```

Object List

```
VARIABLES
Local
> carId: Long@43 "3"
> selectQuery: "SELECT * FROM car WHERE id = ?
> connection: ConnectionImpl@81
  preparedStatement: ClientPreparedStatement
> this: CarService@41
```

Breakpoint List

```
BREAKPOINTS
  Uncaught Exceptions
  Caught Exceptions
  CarController.java   src\main\java\com\carrental\a...  23
  CarController.java   src\main\java\com\carrental\a...  30
  CarController.java   src\main\java\com\carrental\a...  36
  CarService.java      src\main\java\com\carrental\aut...  115
```

Work together

Let's Take a Look.

DAY 9
LESSON 3

Navigating Code with your IDE

NAVIGATING CODE WITH YOUR IDE

REFERENCES

8 results in 3 files

- DatabaseConfig.java src\main\java\com\car...
  - Connection getConnection() throws SQLExc...
- CarService.java src\main\java\com\carrental...
  - DatabaseConfig.getConnection();
  - DatabaseConfig.getConnection();
  - DatabaseConfig.getConnection();
- CustomerService.java src\main\java\com\c...
  - DatabaseConfig.getConnection();
  - DatabaseConfig.getConnection();
  - DatabaseConfig.getConnection();
  - DatabaseConfig.getConnection();

REFERENCES: IMPLEMENTATIONS

4 results in 4 files

- Angora.java src\main\java\com\mains\A...
  - public class Angora extends Cat {
- Cat.java src\main\java\com\mains\Animals
  - public class Cat extends Domestic {
- Dog.java src\main\java\com\mains\Animals
  - public class Dog extends Domestic {
- Persian.java src\main\java\com\mains\Anim...
  - public class Persian extends Cat {

REFERENCES: CLASS HIERARCHY

- Object java.lang
  - Animal com.mains.Animals
    - Domestic com.mains.Animals
      - Cat com.mains.Animals
      - Dog com.mains.Animals

REFERENCES: CALLERS OF

- getConnection() : Connection com.carrental.auto...
- createAndSendReceipt(Long, Long, String) : vo...
- getAllCustomers() : List<CustomerResponseDto>...
- addCar(CarRequestDto) : CarResponseDto com.ca...
- bookCar(BookVehicleDto) : CustomerResponseDt...
- unregisterCarById(Long) : String com.carrental.auto...
- getCustomerById(Long) : CustomerResponseDto...
- getCarById(Long) : CarResponseDto com.carrental....

Lets See it

DAY 8
LESSON 4

Java Reflection

JAVA REFLECTION

Instantiating when class name is not known at development time.

```
String className = "com.mains.Reflection.TheClass";
Class<?> clazz = Class.forName(className);
ClassInterface instance = (ClassInterface) clazz.getDeclaredConstructor().newInstance();
System.out.println("Id = " + instance.getId() + " Name = " + instance.getName() + " Number = " + instance.getNumber());
```

Print Modifiers

```
int modifiers = clazz.getModifiers();
System.out.println("Modifiers: " + Modifier.toString(modifiers));
```

Print Declared Fields

```
Field[] fields = clazz.getDeclaredFields();
for (Field field : fields) {
    System.out.println("  " + field.getType().getSimpleName() + " " + field.getName());
}
```

Print Declared Methods

```
Method[] methods = clazz.getDeclaredMethods();
for (Method method : methods) {
    System.out.print("  " + method.getReturnType().getSimpleName() + " " + method.getName() + "(");
    Class<?>[] parameterTypes = method.getParameterTypes();
    for (int i = 0; i < parameterTypes.length; i++) {
        System.out.print(parameterTypes[i].getSimpleName());
        if (i < parameterTypes.length - 1) {
            System.out.print(", ");
        }
    }
    System.out.println(")");
}
```

Lets See it Running

DAY 9
LESSON 5

Introspection

INTROSPECTION

```java
public static boolean isValidCarRequestDto(CarRequestDto dto) throws IllegalAccessException {
    Field[] fields = dto.getClass().getDeclaredFields();
```

Via Reflection

```java
if (field.getName().equals("year")) {
    int value = field.getInt(dto);
```

Via Reflection

```java
BeanInfo bi = Introspector.getBeanInfo(dto.getClass());
PropertyDescriptor[] propertyDescriptors = bi.getPropertyDescriptors();
Method readMethod = propertyDescriptors[0].getReadMethod();
String name = propertyDescriptors[0].getReadMethod().getName();
Object value = readMethod.invoke(dto);
System.out.println(name + " = " + value);
```

Via Introspection

```java
package com.mains.Introspection;
import java.lang.reflect.Method;

public class IntrospectionMain {
    Run | Debug
    public static void main(String[] args) throws Exception {
        // Get the Class object for the class containing the method
        Class<?> clazz = MyIntrospect.class;

        // Get the Method object using introspection
        Method method = clazz.getDeclaredMethod("printMessage", String.class);

        // Create an instance (or use null for static methods)
        MyIntrospect instance = new MyIntrospect();

        // Invoke the method
        method.invoke(instance, "Hello from introspection!");
    }
}
```

Via Introspection

Lets See it Running

CODING TRIVA QUESTION

In the **IntrospectionMain** class, we just learned how to do introspection, now add reflection, in other words, we want **MyIntrospect** class to be in a string.

You can use **Reflection.ReflectionMain** as guide.

CODING TRIVA ANSWER

```
// Class<?> clazz = MyIntrospect.class;
String className = "com.mains.Introspection.MyIntrospect";
Class<?> clazz = Class.forName(className);

// Get the Method object using introspection
Method method = clazz.getDeclaredMethod("printMessage", String.class, int.class);

// Create an instance (or use null for static methods)
MyIntrospect instance = new MyIntrospect();

// Invoke the method
method.invoke(instance, "Number is = ", 20);
```

Lets See the Code

CODING EXERCISE TO GO

In the **car-controller /api/cars/{carId}** change the line bellow to call method **getCarById(carId)** via introspection. Use a String variable to store **"getCarById"**. Use **introspection.IntrospectionMain** as guide

```
CarResponseDto car = carService.getCarById(carId);
```

GET /api/cars/{carId}

THANK YOU

Crash Course

We will see you Wednesday