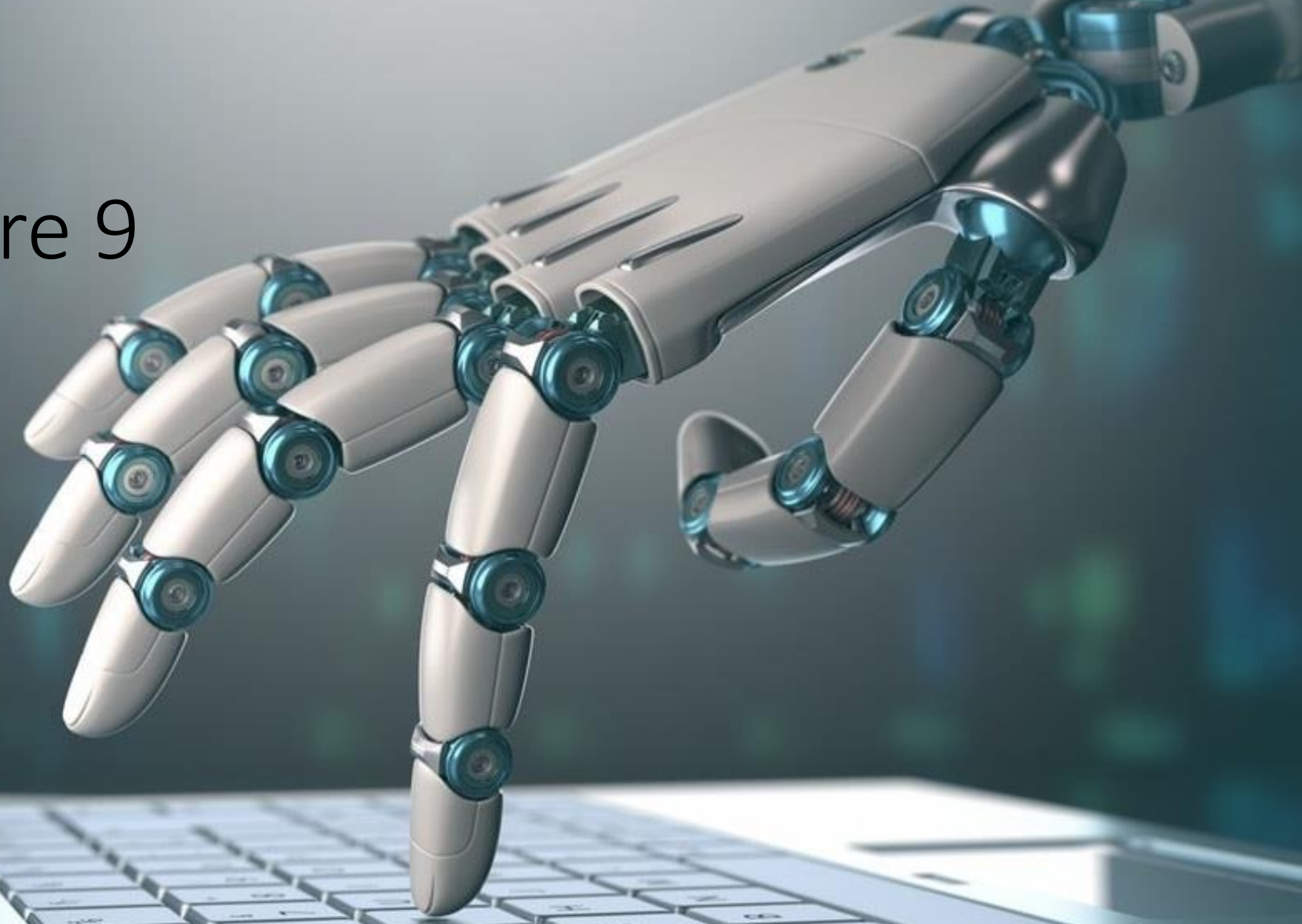


# Lecture 9



# Agenda

- **Big Script**
- Automation project Architecture
- Page Factory
- Model Objects
- Practice

# Big Script

```
public static void main(String[] args) {
    System.out.println("launching chrome browser");
    String driverPath = "C:/Users/Tarkon/IdeaProjects/FinalProjectDemo/src/main/resources/";
    System.setProperty("webdriver.chrome.driver", driverPath + "chromedriver.exe");
    driver = new ChromeDriver();
    driver.navigate().to( url: "http://ss.com/");

    driver.findElement(By.linkText("Vieglie auto")).click();

    driver.findElement(By.name("topt[8][min]")).sendKeys( ...keysToSend: "6000");
    driver.findElement(By.name("topt[8][max]")).sendKeys( ...keysToSend: "10000");

    Select dropdownYear = new Select(driver.findElement(By.name("topt[18][min]")));
    dropdownYear.selectByValue("2009");

    Select dropdownEngine = new Select(driver.findElement(By.name("topt[15][max]")));
    dropdownEngine.selectByValue("3.0");

    Select dropdownColor = new Select(driver.findElement(By.name("opt[17]")));
    dropdownColor.selectByVisibleText("Balta");

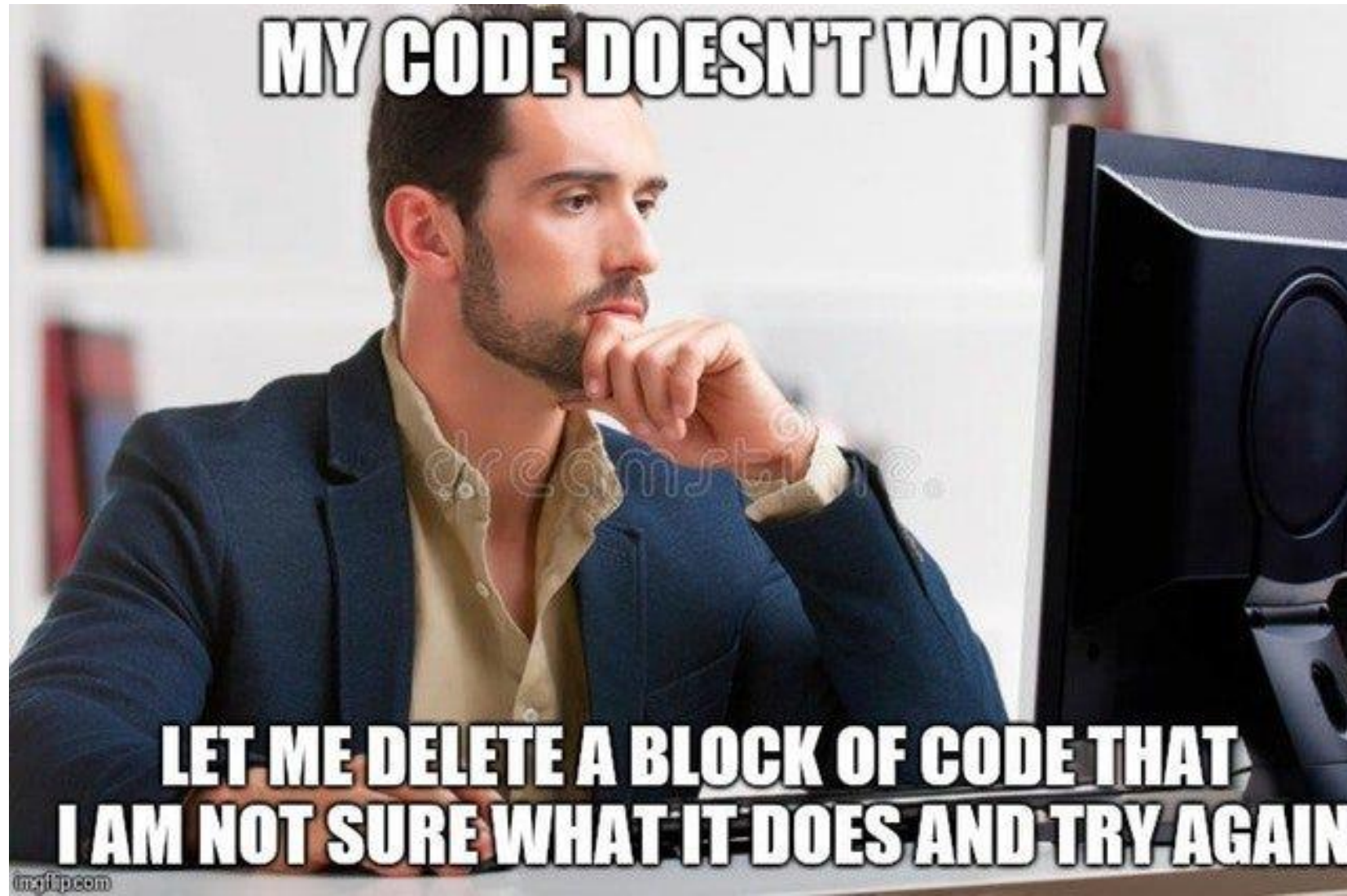
    driver.findElement(By.xpath("//input[@value='Meklēt']")).click();

    try {
        Thread.sleep( millis: 6000 );
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    driver.close();
}
```

# Big Script - why it is bad?

- Not readable
- Not maintainable
- A big mess

So what should we do



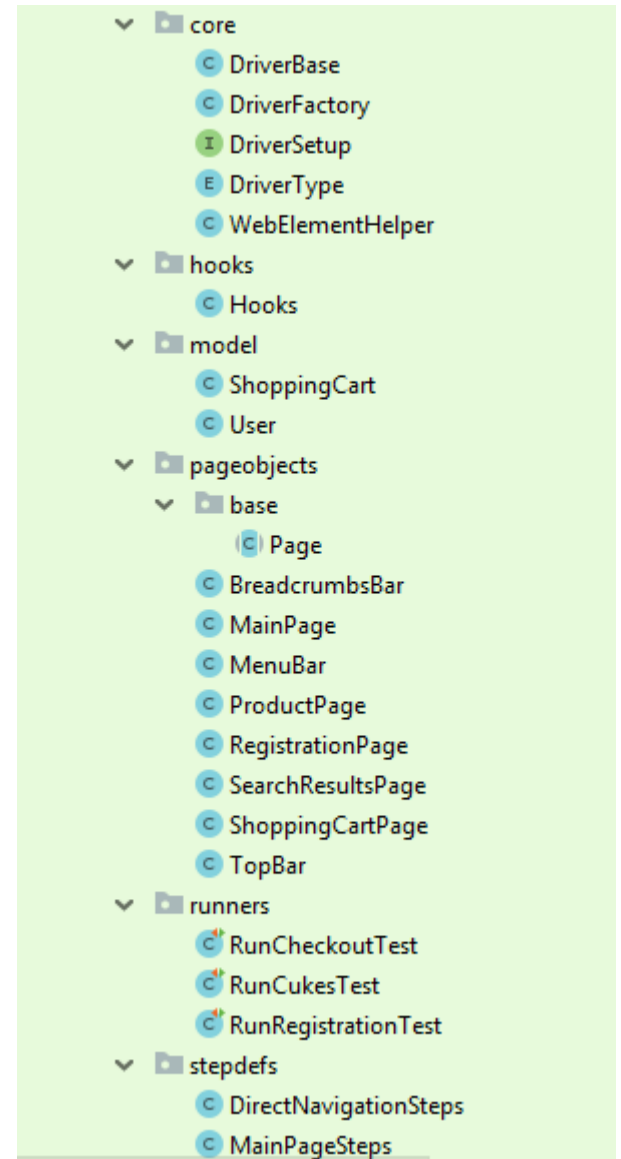
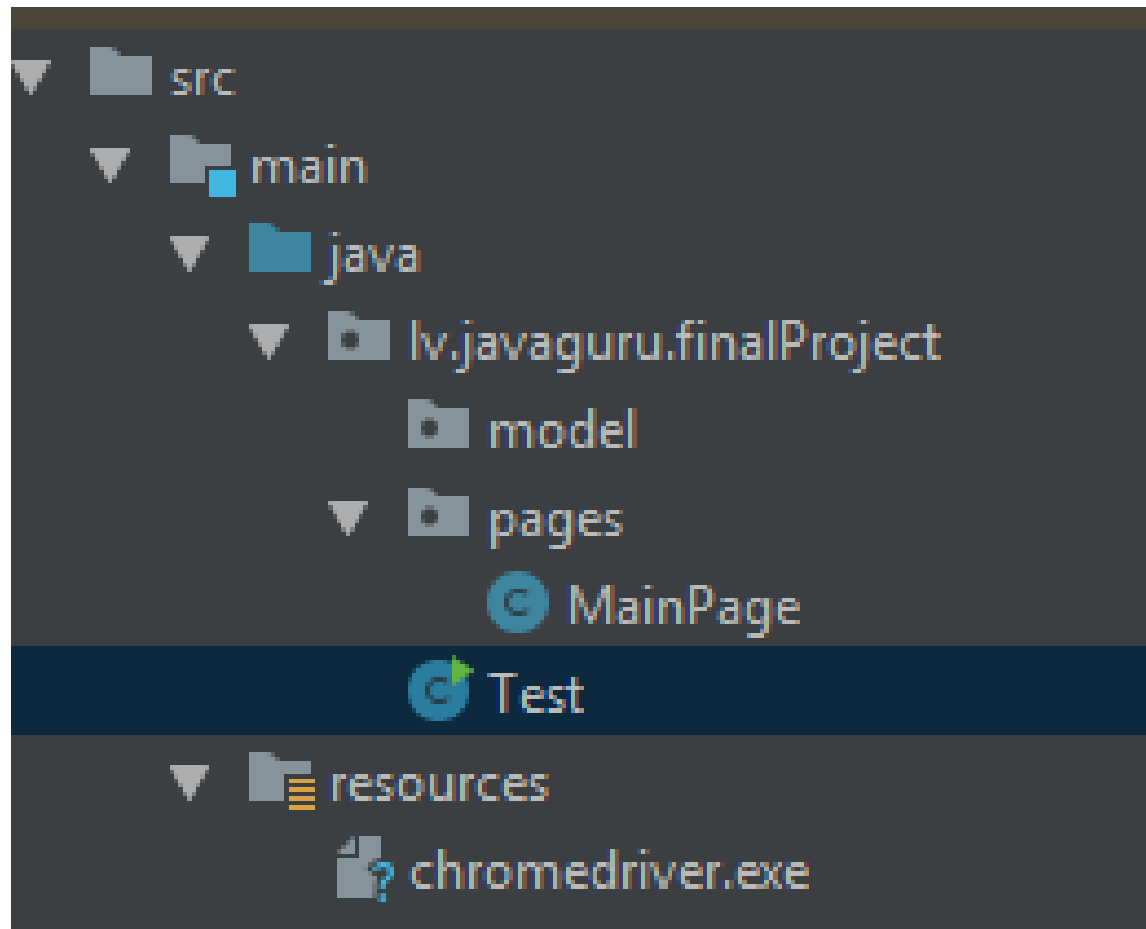
# Agenda

- Big Script
- **Automation project Architecture**
- Page Factory
- Model Objects
- Practice

# Lets begin with architecture (basics)

- Split classes
- Group them
- Proper naming (classes, variables, methods)
- Refactoring

# Automation project Architecture





# Agenda

- Big Script
- Automation project Architecture
- **Page Factory**
- Model Objects
- Practice

# Page Object Factory – More Advance POM

```
public void main() {  
  
    openMyWebPage ( url: "http://ss.com/" );  
    MainPageWithPageFactory elementLocation = PageFactory.initElements(driver, MainPageWithPageFactory.class);  
    elementLocation.navigateToCategory("Vieglie auto");  
    elementLocation.searchByMinAndMaxPrice("6000", "10000");  
    elementLocation.searchByYearMin("2009");  
    elementLocation.serachByEngineMax("3.0");  
    elementLocation.serachByColor("Balta");  
    elementLocation.clickSearchButton();  
  
}
```

# Page Object Factory – More Advance POM

```
@FindBy(name = "topt[8][min]")
private WebElement priceMin;

@FindBy(name = "topt[8][max]")
private WebElement priceMax;

@FindBy(name = "topt[18][min]")
private WebElement yearMin;

@FindBy(name = "topt[15][max]")
private WebElement engineMax;

@FindBy(name = "opt[17]")
private WebElement colorSearch;

@FindBy(xpath = "//input[@value='Meklēt']")
private WebElement searchButton;

public MainPageWithPageFactory (WebDriver driver1) { this.driver = driver1; }

public void navigateToCategory(String category) { driver.findElement(By.linkText(category)).click(); }

public void searchByMinAndMaxPrice(String min, String max) {
    priceMin.sendKeys(min);
    priceMax.sendKeys(max);
}

mainPageWithPageFactory > clickSearchButton()
```

# Page Factory – another example

```
1 //*****Web Elements by using Page Factory*****
2 @FindBy(how = How.ID, using = "email")
3 public WebElement username;
4
5 @FindBy(how = How.ID, using = "password")
6 public WebElement password;
7
8 @FindBy(how = How.ID, using = "loginButton")
9 public WebElement loginButton;
10
11 @FindBy(how = How.XPATH, using = "//*[@id=\"loginForm\"]/div[1]/div/div")
12 public WebElement errorMessageUsername;
13
14 @FindBy(how = How.XPATH, using = "//*[@id=\"loginForm\"]/div[2]/div/div")
15 public WebElement errorMessagePassword;
```

# BUT

- **Simon Stewart**, **Selenium** project **lead** and **creator** of the Page Factory, at the **2017 SeleniumConf** in Austin. During his keynote address he says not to use Page Factory. This section of the talk starts here:
- <https://youtu.be/gyfUpOysIF8?t=1517>
- Actual statement is at 27:25.

# My Personal Opinion and Some other automation guys opinion

- It is ok to use Page Factory
- No difference from POM
- Just opinion of choice
- ***BUT Please do NOT USE it – problems with debugging, work slower for huge projects, create much more problems and issues in big problems due initiation for each page***

# Agenda

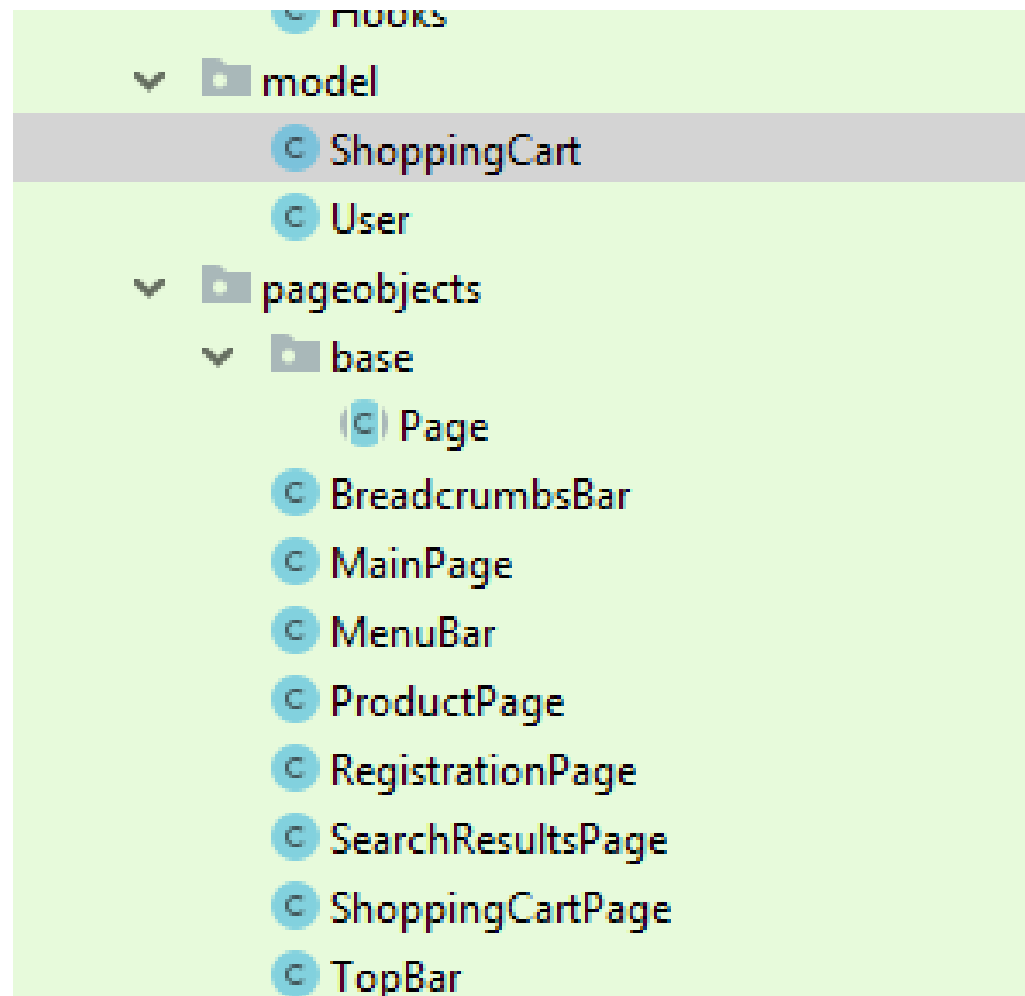
- Big Script
- Automation project Architecture
- Page Factory
- **Model Objects**
- Practice

# What is model?

- Model – simple getter and setter object which we use for validation
- Setup in the beginning of test and validating in the end



# Create models



```
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class User {
    String firstName;
    String lastName;
    String email;
}
```

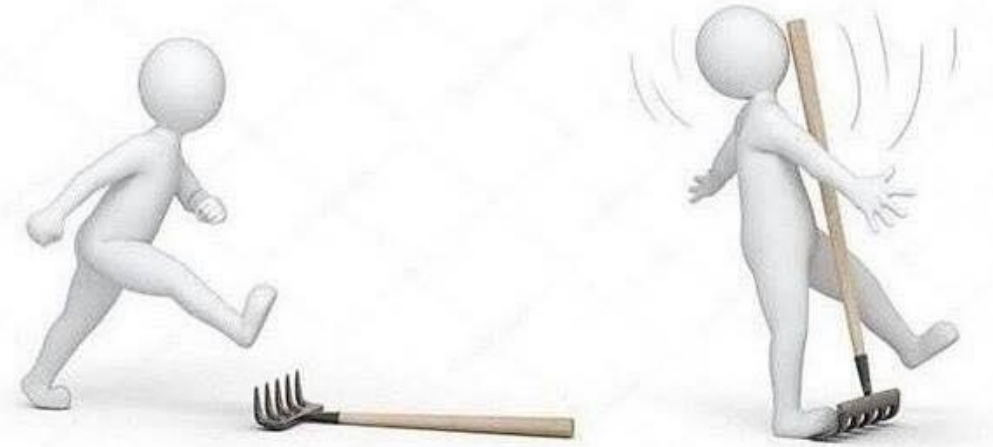
# Create models

```
public class CarModel {  
  
    private String color;  
    private int number;  
    private int year;  
  
    public String getColor() { return color; }  
  
    public void setColor(String color) { this.color = color; }  
  
    public int getNumber() { return number; }  
  
    public void setNumber(int number) { this.number = number; }  
  
    public int getYear() { return year; }  
  
    public void setYear(int year) { this.year = year; }  
  
    public String getPlate() { return plate; }  
  
    public void setPlate(String plate) { this.plate = plate; }  
  
    private String plate;  
}
```

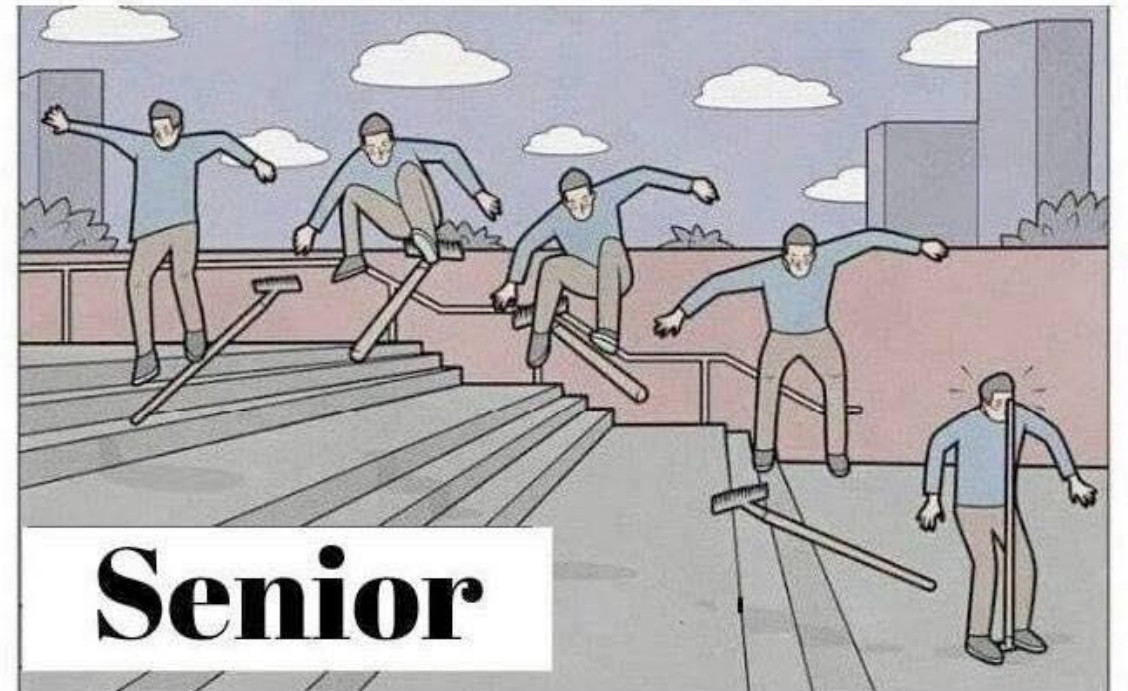
# Agenda

- Big Script
- Automation project Architecture
- Page Factory
- Model Objects
- **Practice**

Why so much practice?



**Junior**



**Senior**

# Task 1

- Open <https://www.janisroze.lv>
- Register account
- Create model where you get all Login data – username and password
- Write test 1 – Positive test case where user successfully logs in to Web – assert that user has logged in to Portal
- Make login method generalized so it can be reused further in tests

## Task 2

- Open <https://www.janisroze.lv>
- Write test 2 – Negative test case where user enters incorrect data – validation for error message – user random info to fill in – Use login method that you have previously used

# Task 3

- Open <https://www.janisroze.lv>
- Write test 3 – Login using one of social network authentication – assert that user has logged in to Portal

# Task 4

- Open <https://www.janisroze.lv>
- In search enter bvs superman
- Click on first found search (assert figure name is Figūra POP! DC: BvS: Superman)
- Get item to basket and go to Basket
- Create a Order model
- Open Order form and fill it
- Enter Coupon name WhyNot?
- DO NOT PRESS order button



# Reference

- Page factory - <https://www.guru99.com/page-object-model-pom-page-factory-in-selenium-ultimate-guide.html>
- Clean code - <https://dzone.com/articles/clean-code-summary-and-key-points>
- Hooks - <https://cucumber.io/docs/cucumber/api/>



**THANK YOU FOR YOUR ATTENTION**

