

Selenium for Java Developers

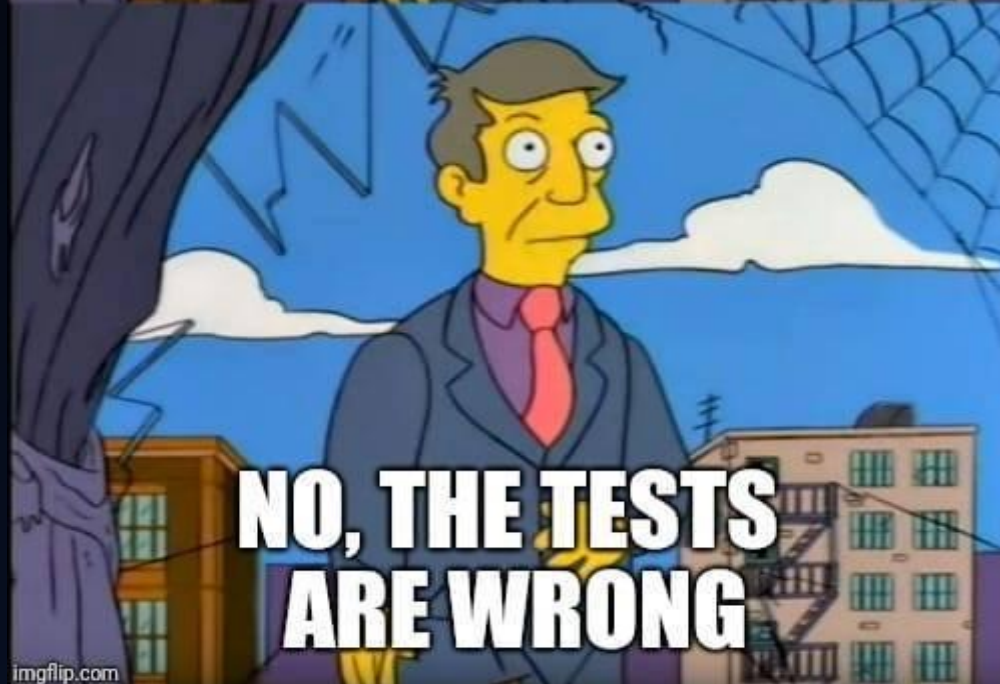
JOHN POURDANIS

Test Automation Engineer

IS MY CODE BAD?

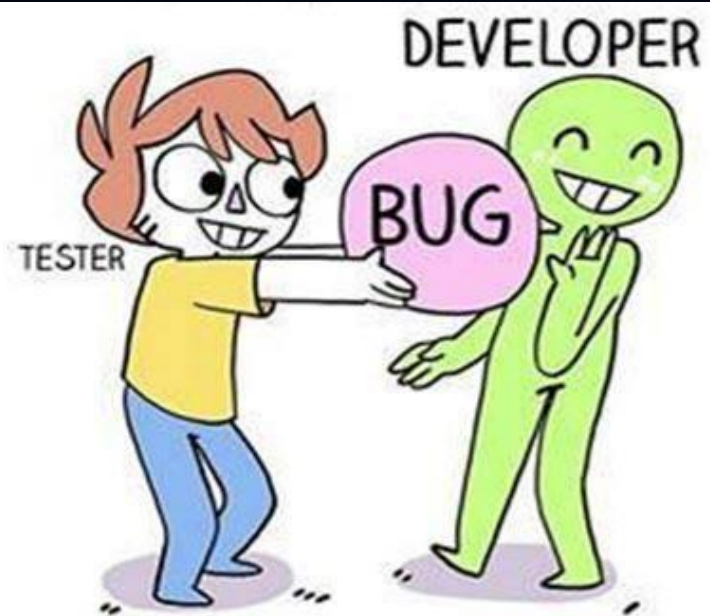


**NO, THE TESTS
ARE WRONG**

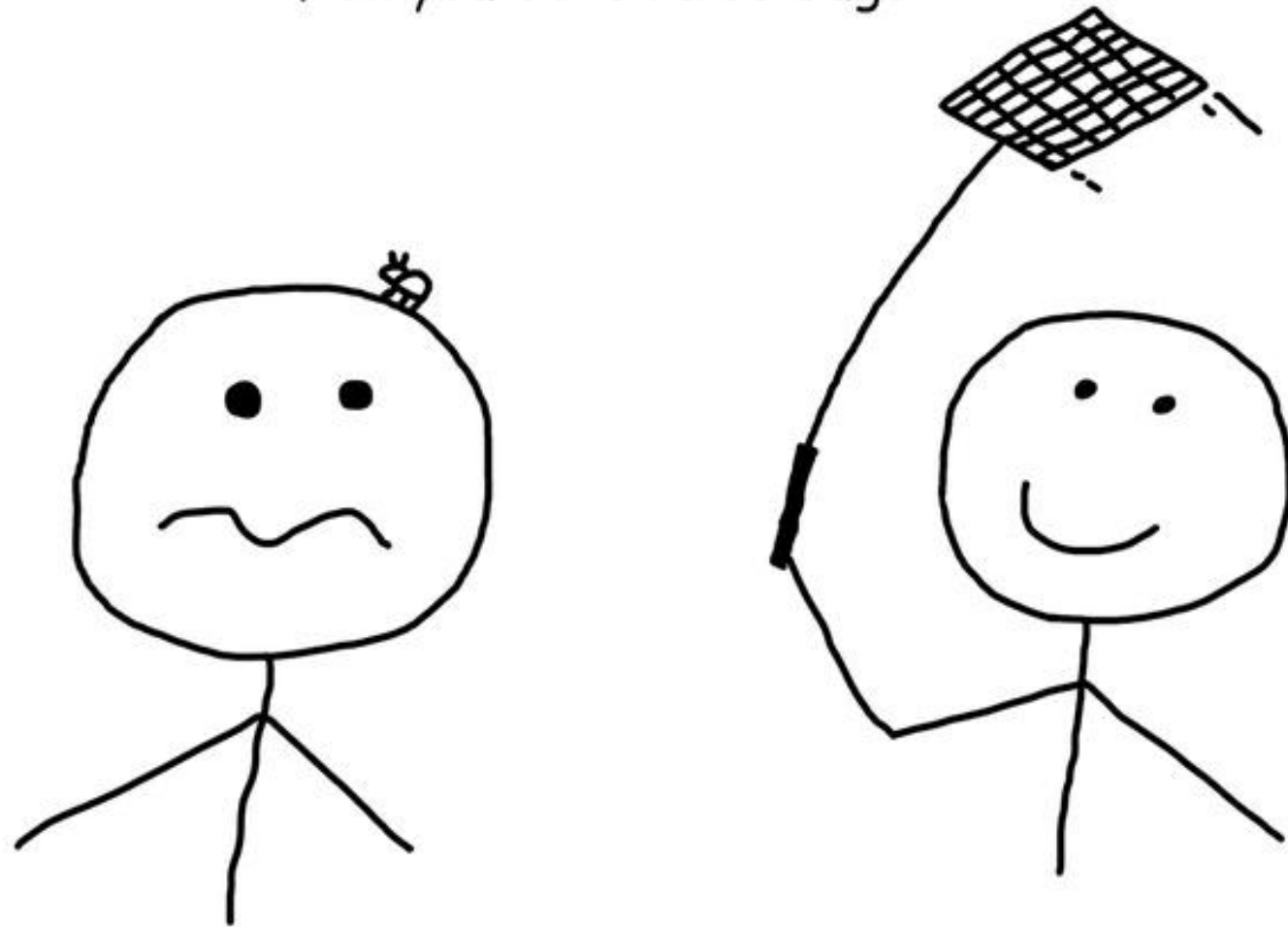


A BUG IN MY CODE?

BITCH, ITS A FEATURE

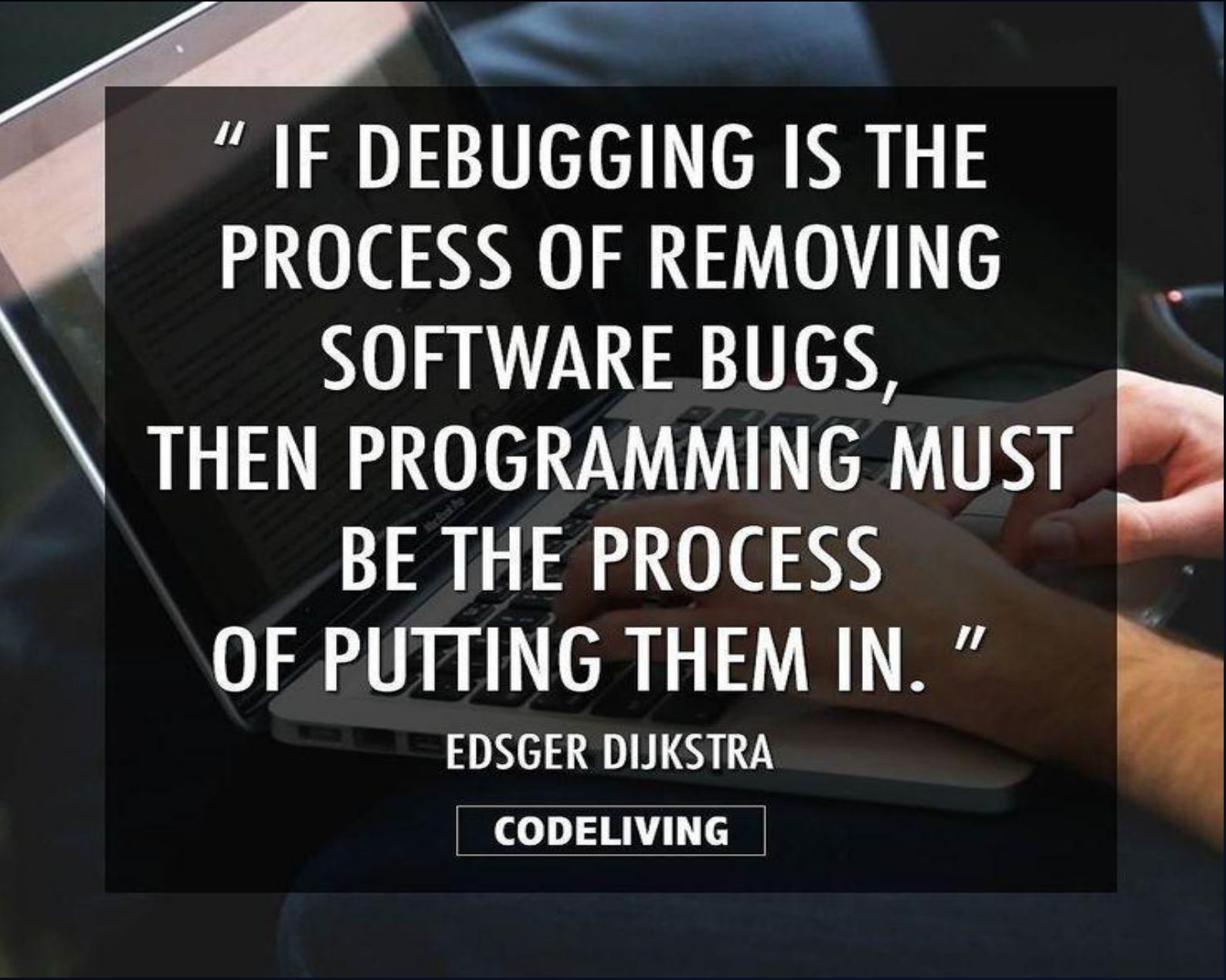


How a developer might feel
if all you do is raise bugs



"Ah, there's the bug! Don't... move... one... bit"

AG

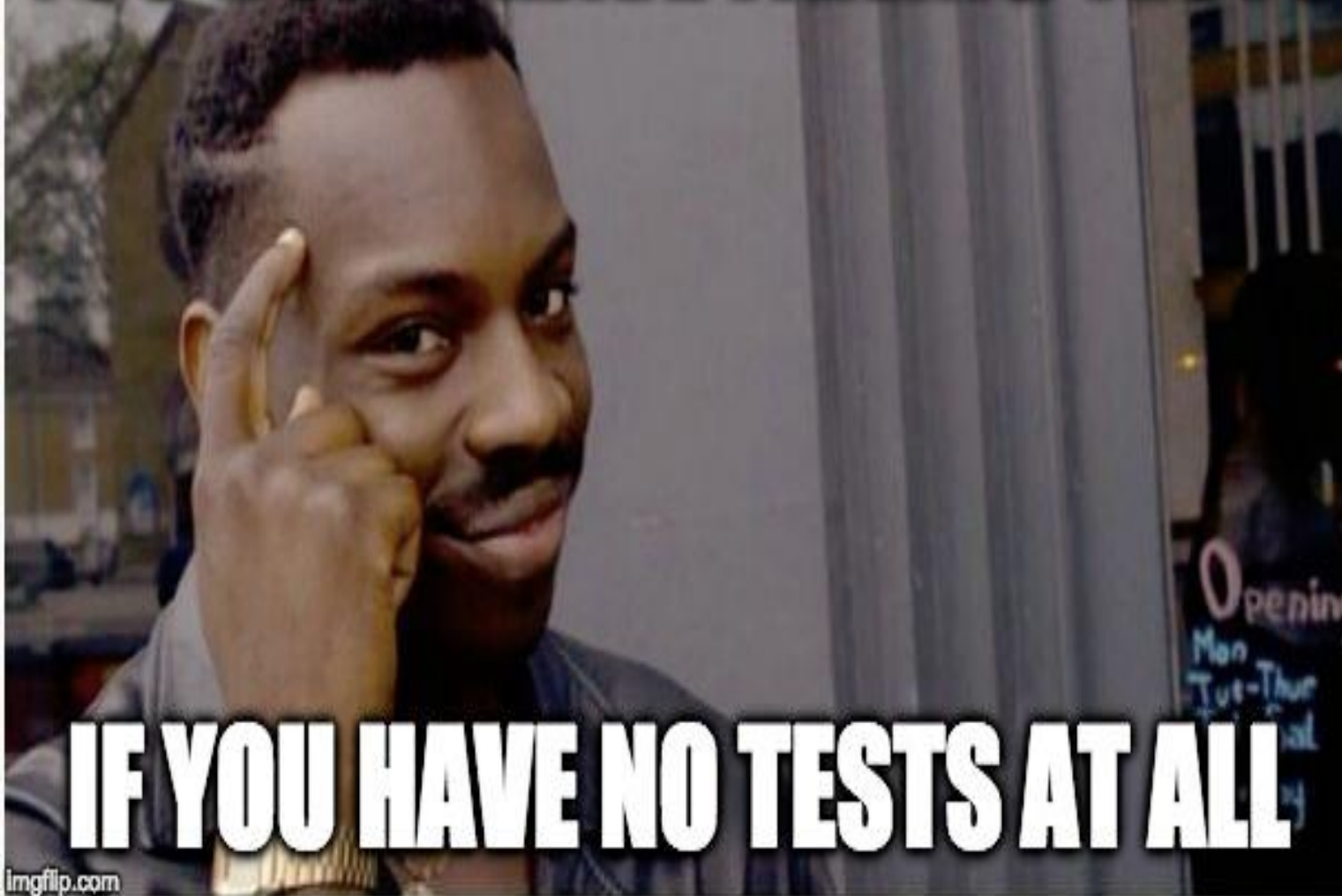
A close-up photograph of a person's hands typing on a laptop keyboard. The image is slightly blurred, focusing on the text overlay. The background is a dark, moody scene of a laptop and hands. On the far left, there are several thin, parallel teal lines that extend vertically.

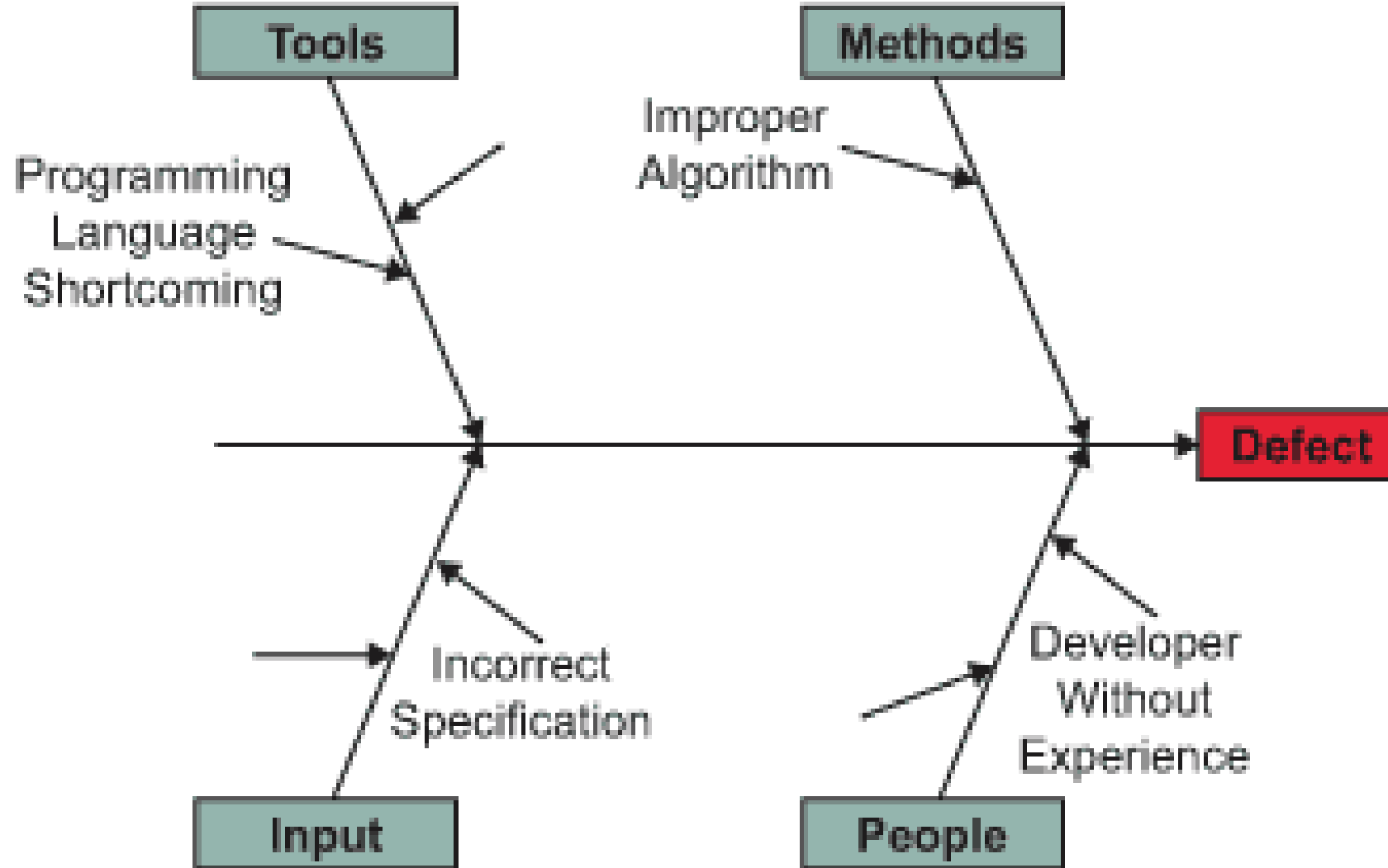
**" IF DEBUGGING IS THE
PROCESS OF REMOVING
SOFTWARE BUGS,
THEN PROGRAMMING MUST
BE THE PROCESS
OF PUTTING THEM IN. "**

EDSGER DIJKSTRA

CODELIVING

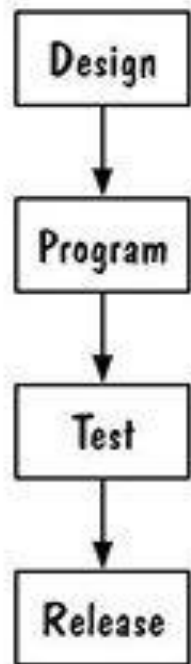
YOU DON'T HAVE FAILING TESTS



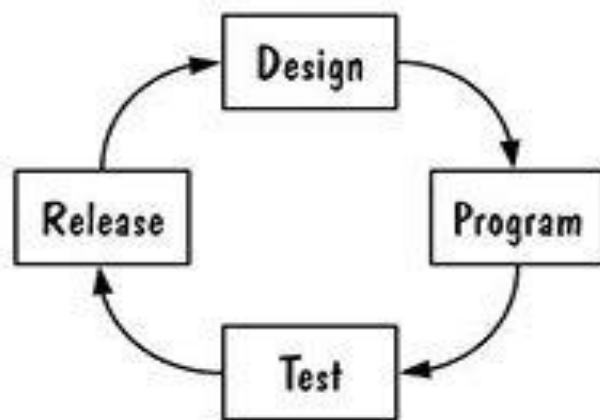




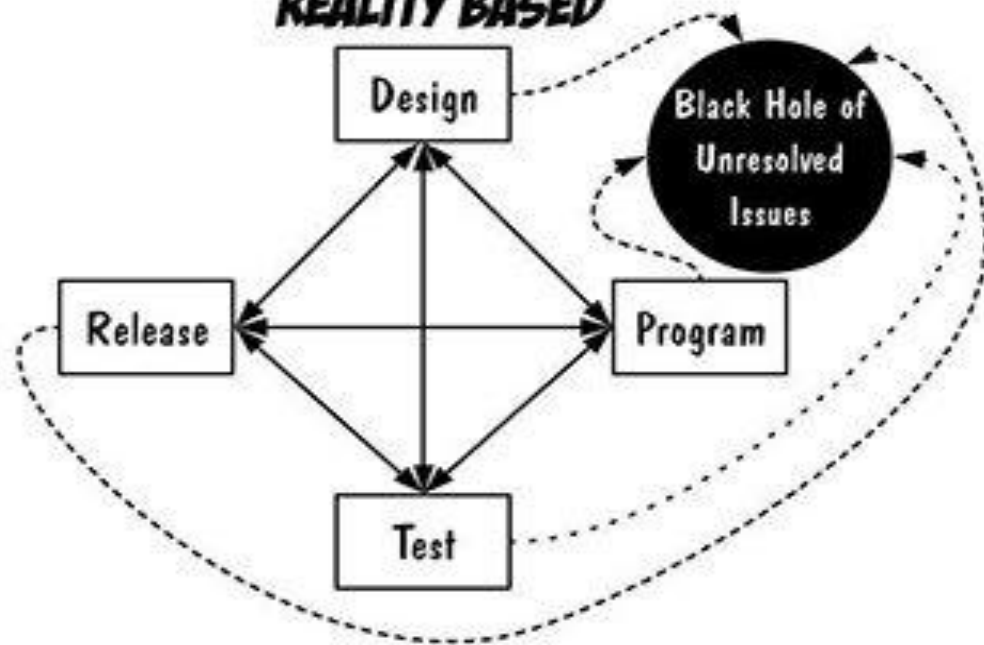
WATERFALL



AGILE



REALITY BASED

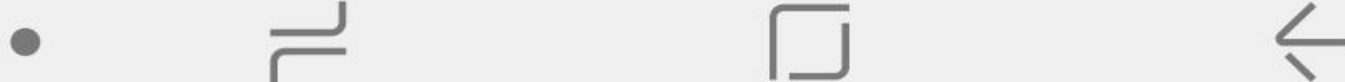


Reshob 2018

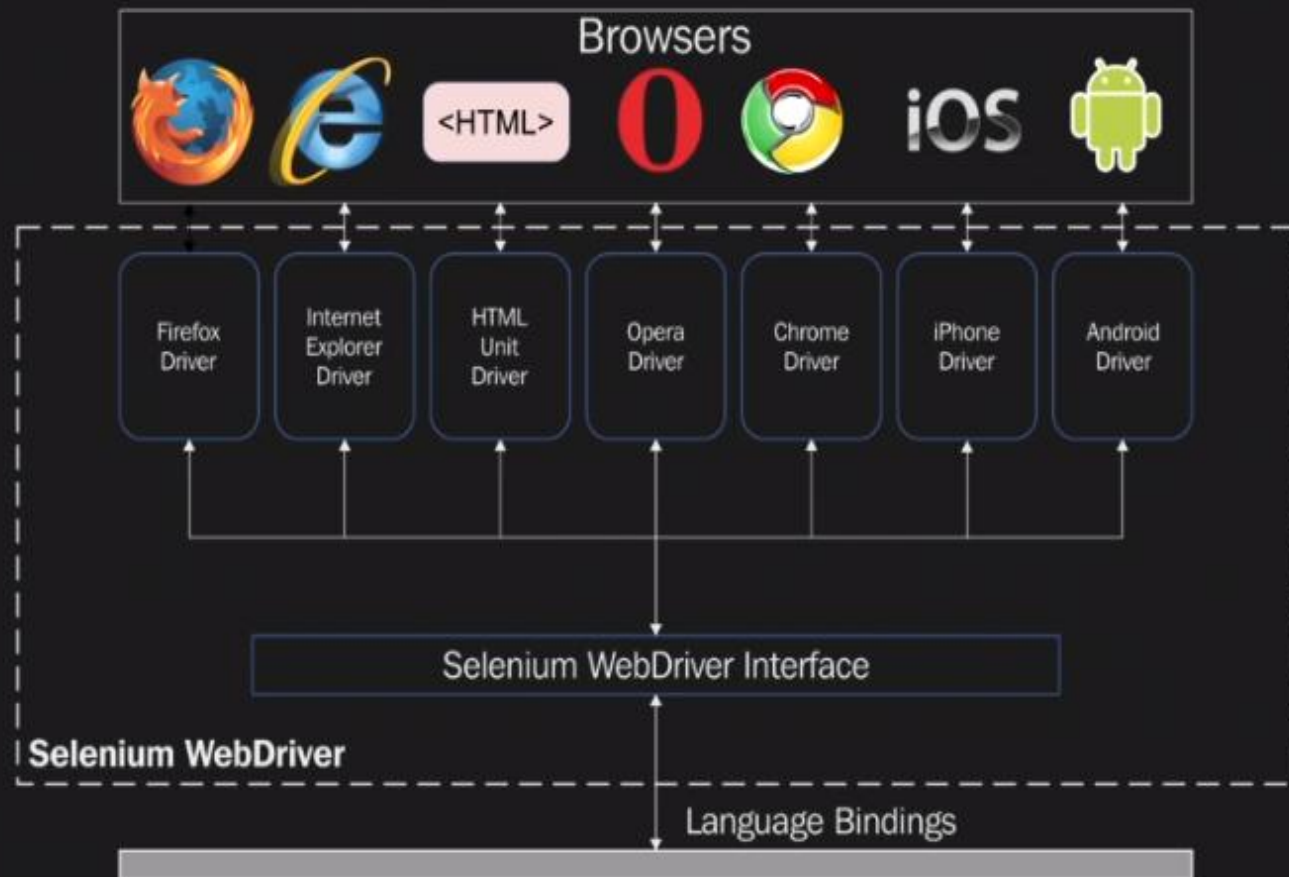
DESIRED

- Experience with programming in Java, C++ and C; Python.
- Scrum certification
- Experience with testing mythologies
- Experience with versions control systems, preferably Subversion.
- Experience with industrial or autonomous robots.

Job Details



Architectural Overview...



```
public class FirstTest {

    //-----Global Variables-----
    //Declare a Webdriver variable
    public WebDriver driver;

    //Declare a test URL variable
    public String testURL = "https://www.meetup.com/Thessaloniki-Java-Meetup-Group/";

    //-----Test Setup-----
    @BeforeMethod
    public void setupTest () {
        //Create a new ChromeDriver
        driver = new ChromeDriver();

        //Go to www.swtestacademy.com
        driver.navigate().to(testURL);
    }

    //-----Tests-----
    @Test
    public void firstTest () {
        //Get page title
        String title = driver.getTitle();

        //Print page's title
        System.out.println("Page Title: " + title);

        //Assertion
        Assert.assertEquals(title, "Thessaloniki Java Meetup Group", "Title assertion is failed!");
    }

    //-----Test TearDown-----
    @AfterMethod
    public void teardownTest () {
        //Close browser and end the session
        driver.quit();
    }
}
```


Abstraction

*"Eliminate the Irrelevant,
Amplify the Essential"*

Encapsulation

"Hiding the Unnecessary"

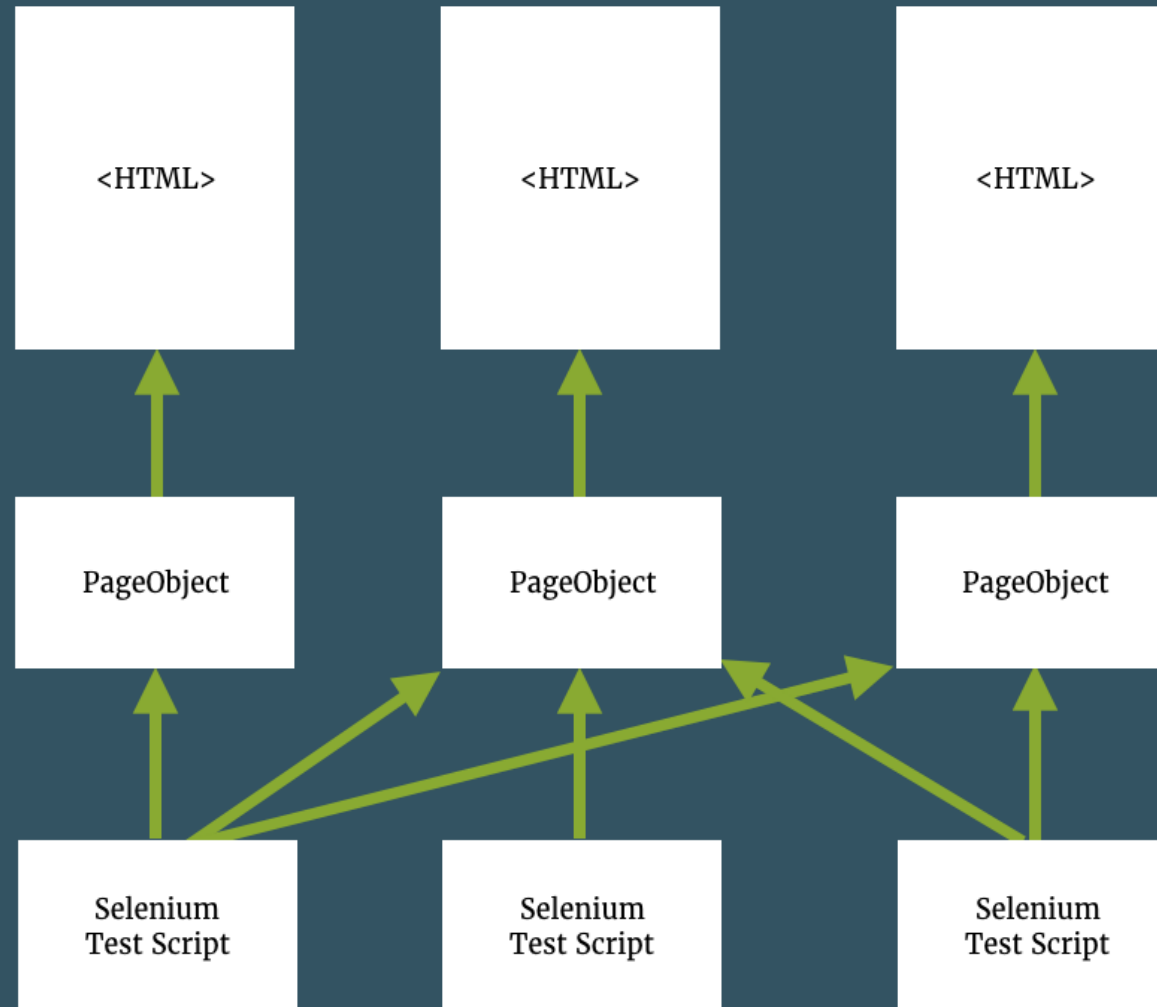
Inheritance

"Modeling the Similarity"

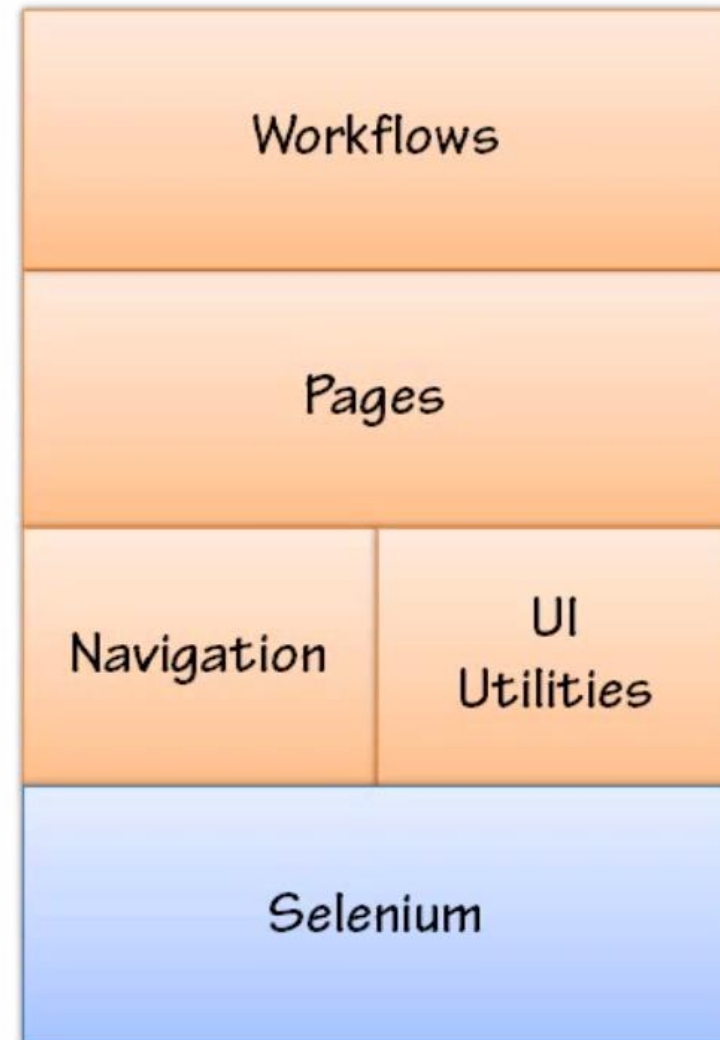
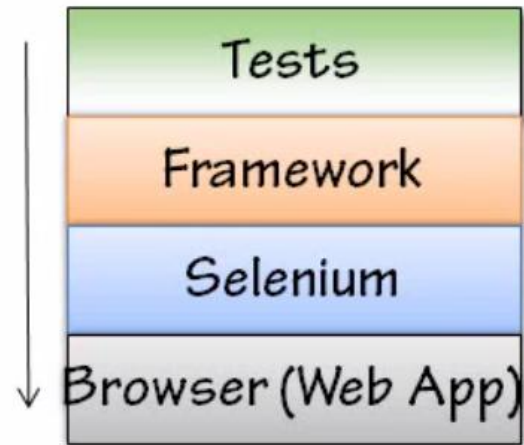
Polymorphism

"Same Function Different Behavior"

Abstraction



Thinking About Layers



Do you need a layer here to abstract away a particular automation tool?

Abstraction

```
public class MeetupPage {  
  
    protected WebDriver driver;  
  
    // Locators of html elements  
    private By membersLink = By.Id("MemberLink");  
    private By pastMeetupsLink = By.Name("meetup");  
    private By photosLink = By.Xpath("//*[@id=muMain]/d/span");  
  
    // Constructor of page object  
    public MeetupPage (WebDriver driver){  
        this.driver = driver;  
    }  
  
    // Action that can on this page  
    public void open(String url){  
        driver.get(url);  
    }  
  
    public void close(){  
        driver.quit();  
    }  
  
    public String getTitle(){  
        return driver.getTitle();  
    }  
  
    public MembersPage navigateToMemberPage(){  
        driver.findElement(membersLink).click();  
        return new MembersPage(this.driver);  
    }  
  
    public PastMeetupPage navigateToPastMeetupPage(){  
        driver.findElement(pastMeetupsLink).click();  
        return new PastMeetupPage(this.driver);  
    }  
  
    public PhotosPage navigateToPhotosPage(){  
        driver.findElement(photosLink).click();  
        return new PhotosPage(this.driver);  
    }  
}
```


More Abstraction

```
public class Page {

    protected WebDriver driver;

    // Locators of html elements
    private By membersLink = By.Id("MemberLink");
    private By pastMeetupsLink = By.Name("meetup");
    private By photosLink = By.Xpath("//*[@id=muMain]/d/span");

    // Constractor of page object
    public Page (WebDriver driver){
        this.driver = driver;
    }

    // Action that can on this page
    public void open(String url){
        driver.get(url);
    }

    public void close(){
        driver.quit();
    }

    public String getTitle(){
        return driver.getTitle();
    }

    public MembersPage navigateToMemberPage(){
        driver.findElement(membersLink).click();
        return new MembersPage(this.driver);
    }

    public PastMeetupPage navigateToPastMeetupPage(){
        driver.findElement(pastMeetupsLink).click();
        return new PastMeetupPage(this.driver);
    }

    public PhotosPage navigateToPhotosPage(){
        driver.findElement(photosLink).click();
        return new PhotosPage(this.driver);
    }

}
```

Inheritance

```
public class MyMeetupPage extends Page {

    protected WebDriver driver;

    // Locators of html elements
    private By manageGroupButton = By.Id("manageGroupbtn");
    private By contactMemberLink = By.Id("contact");
    private By planMeetupButton = By.Name("plan");

    // Constractor of page object
    public MyMeetupPage (WebDriver driver){
        this.driver = driver;
    }

    // Action that can on this page
    public void manageGroup(){
        driver.findElement(manageGroupButton).click();
        driver.findElement(contactMemberLink).click();
    }
}
```

Encapsulation

```
public class LoginPage {  
  
    private IWebDriver driver;  
  
    private final static String username = "TestUser_1";  
    private final static String password = "Test@123";  
  
    [FindBy(How = How.Id, Using = "log")]  
    private IWebElement UserName { get; set; }  
  
    [FindBy(How = How.Id, Using = "pwd")]  
    private IWebElement Password { get; set; }  
  
    [FindBy(How = How.Id, Using = "login")]  
    private IWebElement Submit { get; set; }  
  
    public LoginPage(IWebDriver driver){  
        this.driver = driver;  
        PageFactory.InitElements(driver, this);  
    }  
  
    public void LoginToApplication(){  
        UserName.SendKeys(username);  
        Password.SendKeys(password);  
        Submit.Click();  
    }  
}
```

Polymorphism

```
public Enum LoginMethod{
    Facebook,Google
}

public class LoginPage {

    private IWebDriver driver;

    private
    [FindsBy(How = How.Id, Using = "log")]
    private IWebElement UserName { get; set; }

    [FindsBy(How = How.Id, Using = "pwd")]
    private IWebElement Password { get; set; }

    [FindsBy(How = How.Id, Using = "login")]
    private IWebElement Submit { get; set; }

    public LoginPage(IWebDriver driver){
        this.driver = driver;
        PageFactory.InitElements(driver, this);
    }

    public void LoginToApplication(){
        UserName.SendKeys("TestUser_1");
        Password.SendKeys("Test@123");
        Submit.Click();
    }

    public void LoginToApplication(String username, String password, LoginMethod loginMethod){
        switch (loginMethod) {
            case Facebook:
                LoginWithFacebook(String username, String Password);
                break;
            case Google:
                LoginWithGoogle(String username, String Password);
                break;
        }
    }
}
```