



**DataArt**



QA Automation

Locators. Requirements. Environment.

---

New York USA  
London UK  
Munich Germany  
Zug Switzerland

Anton Sirota



# HTML

---

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

# HTML Tags

---

Page base (Tags: <html>, <head>, <body>)

Page title (Tags: <title>)

Headers (Tags: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>)

Paragraphs (Tags: <p>)

Image (Tags: <img>)

Line brake (Tags: <br />)

Horizontal line (Tags: <hr />)

Hyperlink (Tags: <a>)

Lists (Tags: <ul>, <ol>, <li>)

Containers (Tags: <div>, <span>)

Table (Tags: <table>, <tr>, <td>, <th>)



# Locator Types

---

## Simple

Id  
Name  
Class  
LinkText  
PartialLinkText  
TagName

## Complex

CSS  
XPath

# Simple Locators



---

Easy to write  
Support  
Performance



# Complex Locators

---

When there is no way to use simple  
Item need to be searched by position (table)  
Search by multiple item attributes  
Search relative to other items

# CSS Selectors

---

**\***– find any element

**button**– find all elements with tag button

**#myid**– find element with id=myid

**.myclass**– find element with class=myclass

**div[attribute='value']**–find element div with an attribute 'value'

**div button**– find a child of a button inside a div at any nesting level

More here:

[http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)

# Xpath

---

**/** - move to level 1

**//** - transition to several levels

**//\*** - search for any item at any level

**// div** - search div at any level

**/..** – rise to level

**//div/button** – find the button next to the div

**//button[2]** - find the second buttons (following the button)

# Xpath

---

**`//*[@id= 'myid']`** - find an element on the page with id = myid

**`//*[text()='some']`** - find an element with some text

**`//*[contains(@id, '123')]`** - find the element whose id contains 123

**`//*[@name!='Bob']`** - find the element with the name NOT Bob

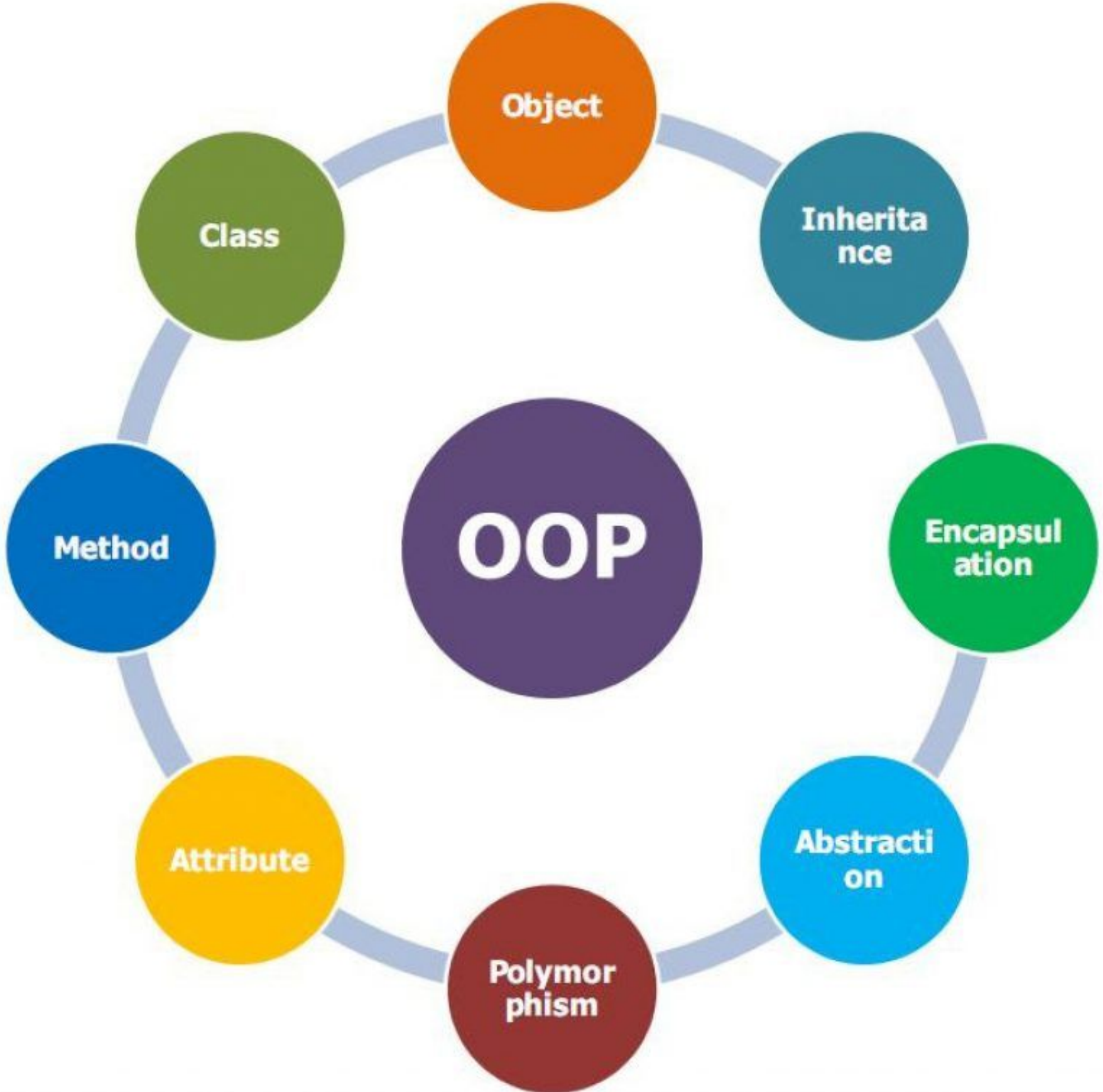
**`//*[@class='a 'and @name='b']`** - example AND

**`//*[@class='a' or @name='b']`** - OR example

**[http://www.w3schools.com/xsl/xpath\\_syntax.asp](http://www.w3schools.com/xsl/xpath_syntax.asp)**

# OOP Principles

# OOP Principles



# OOP Principles

---

## 1. Encapsulation

## 2. Abstraction

## 3. Inheritance

## 4. Polymorphism

```
public class Employee {  
    private String name;  
    private Date dob;  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public Date getDob() {  
        return dob;  
    }  
    public void setDob(Date dob) {  
        this.dob = dob;  
    }  
}
```

# OOP Principles

---



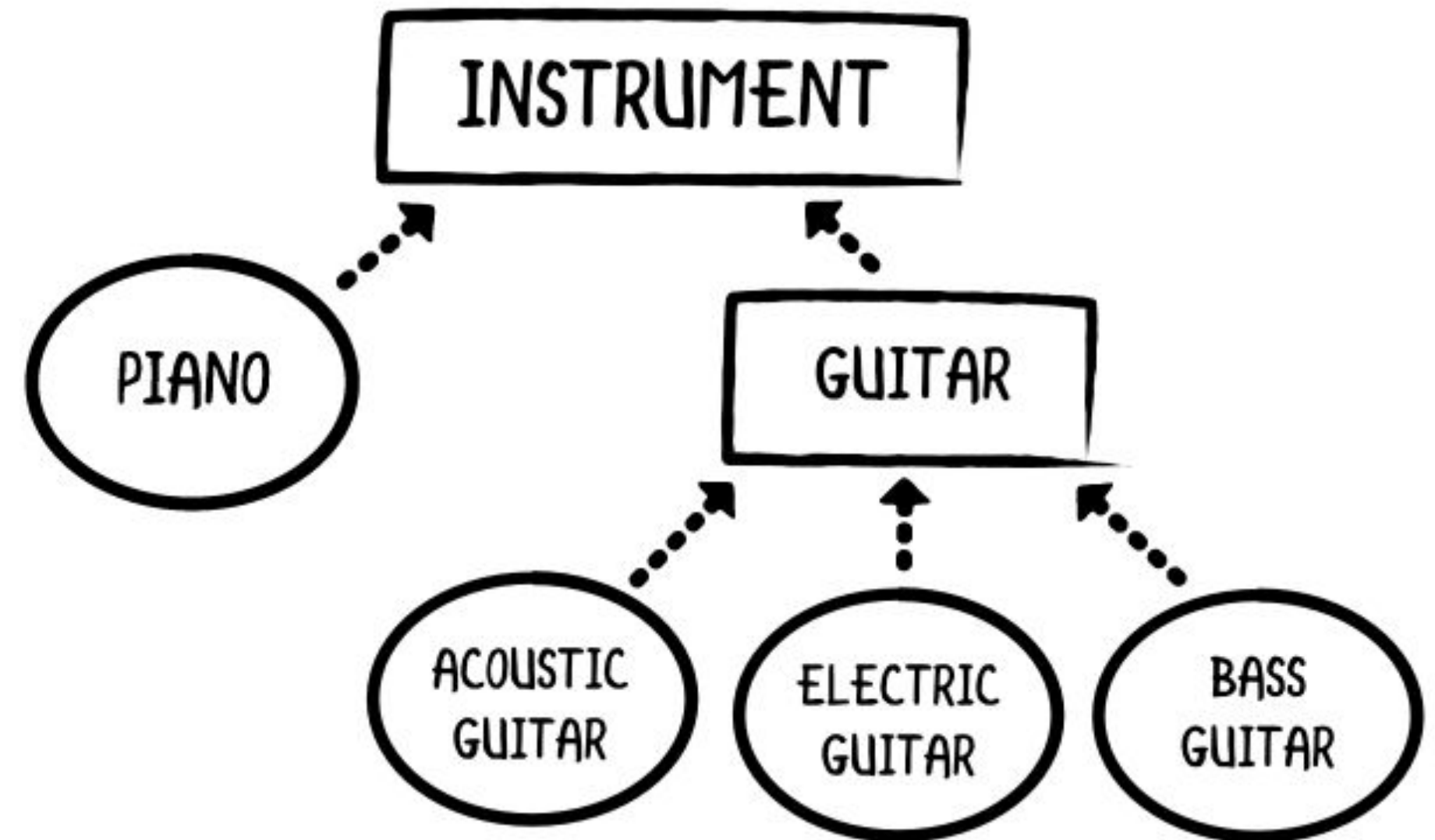
1. Encapsulation
- 2. Abstraction**
3. Inheritance
4. Polymorphism

```
public abstract class Employee {  
    private String name;  
    private String address;  
    private int number;  
  
    public abstract double  
computePay();  
    // Remainder of class definition  
}
```



# OOP Principles

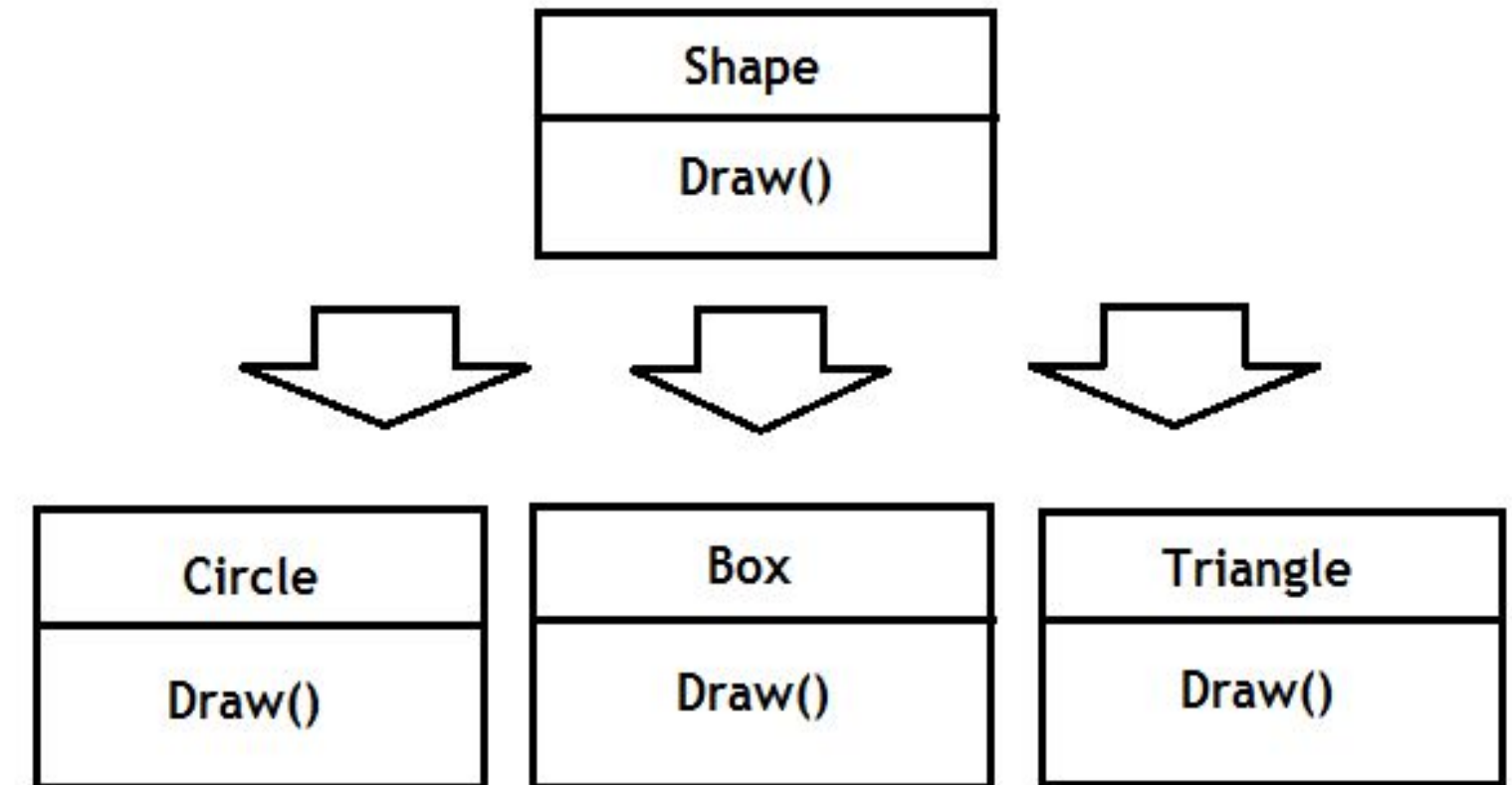
- 1. Encapsulation
- 2. Abstraction
- 3. Inheritance**
- 4. Polymorphism



# OOP Principles

---

1. Encapsulation
2. Abstraction
3. Inheritance
4. **Polymorphism**





# Java Basics

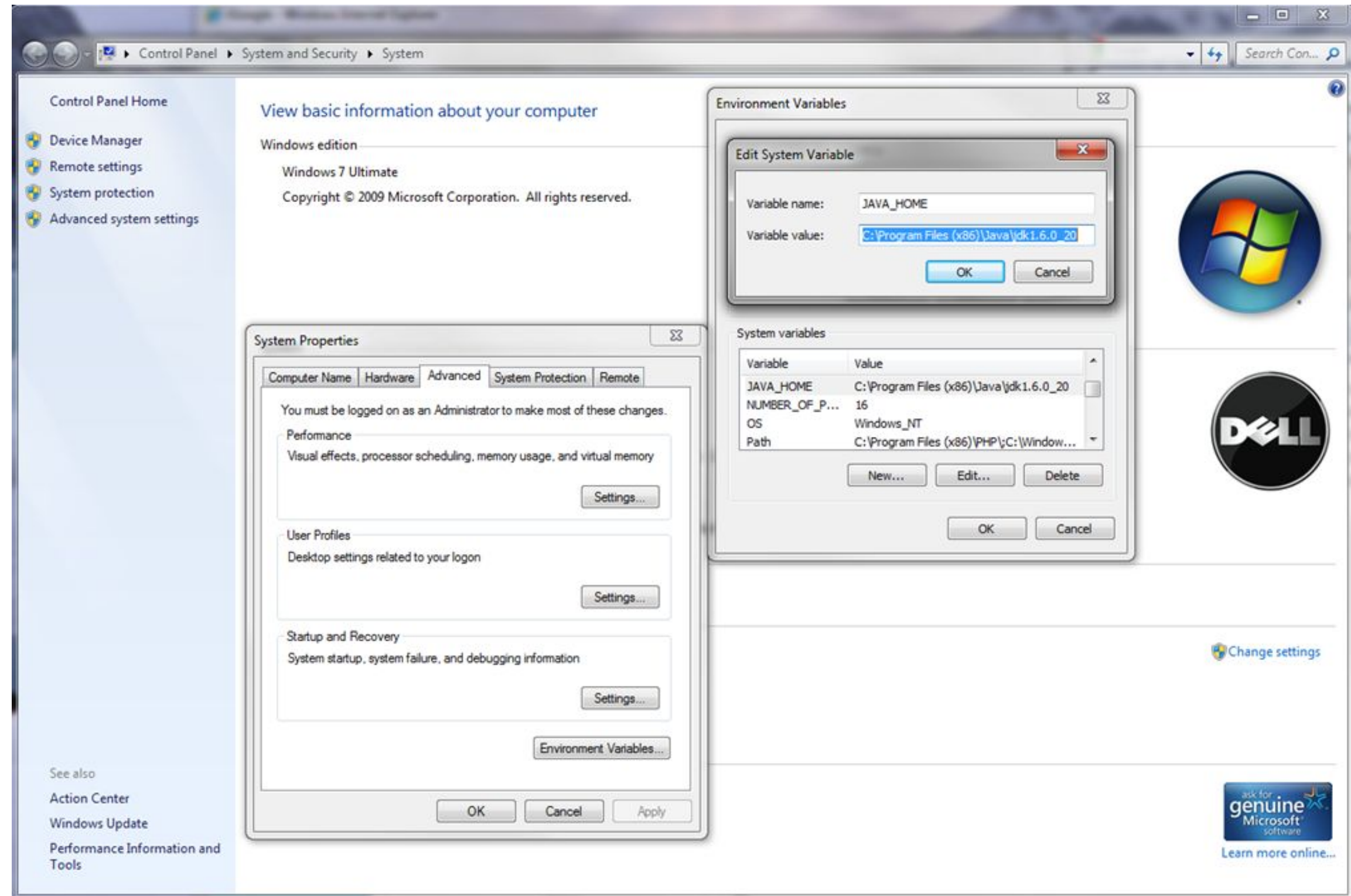
---

- Variables and methods
- Operators and data types (int, double, boolean, Strings etc)
- Modifiers (private, protected...; final, static, abstract)
- Loops (for, while, do-while)
- If-else, switch-case statements
- Collections (ArrayList, HashMap, HashSet etc)
- Exceptions (try/catch/throw/finally)
- Stream Filters with Lambda Expressions



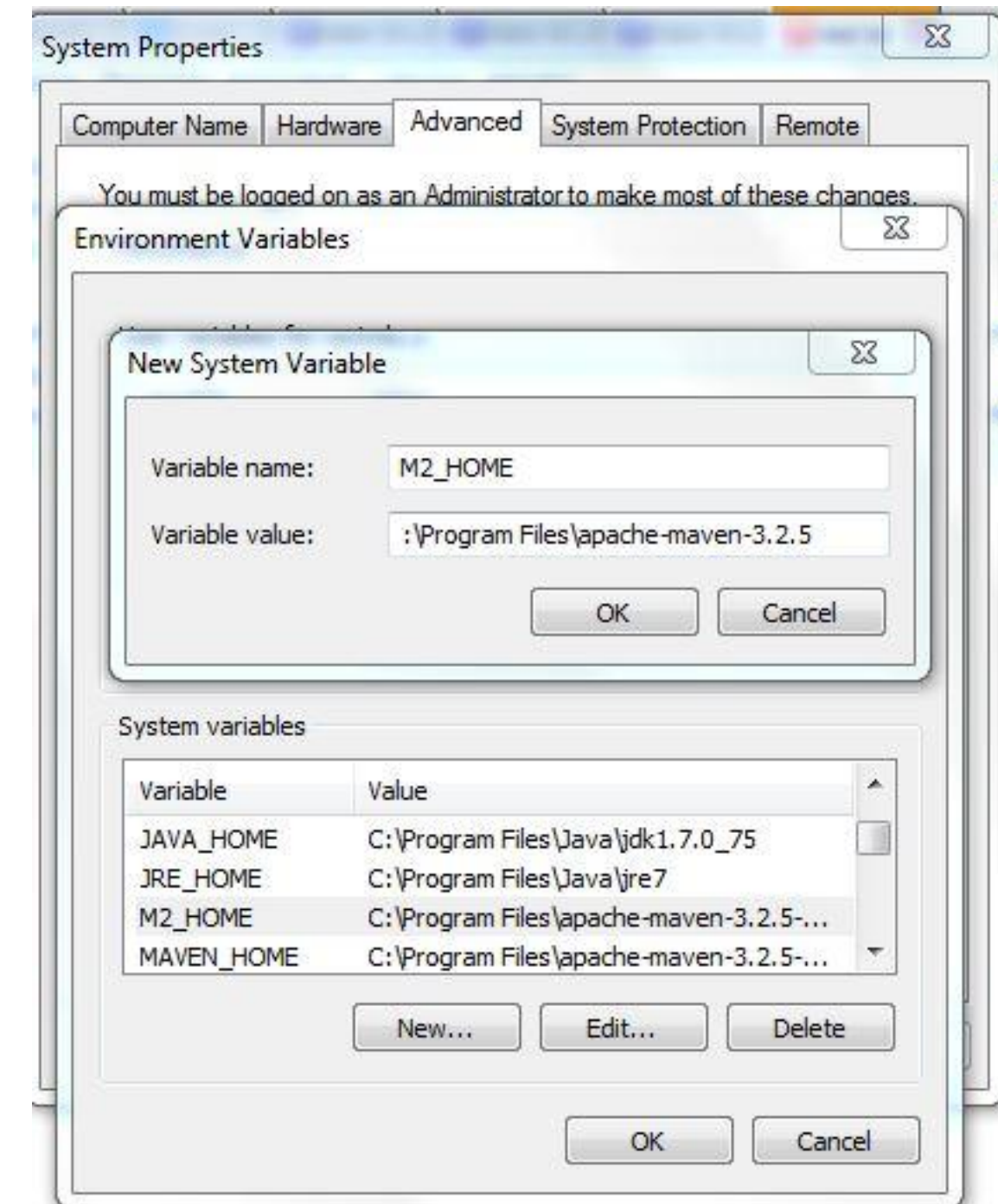
# Java

- Download and install Java Development Kit 8+
- Set JAVA\_HOME as environment variable (see pic).



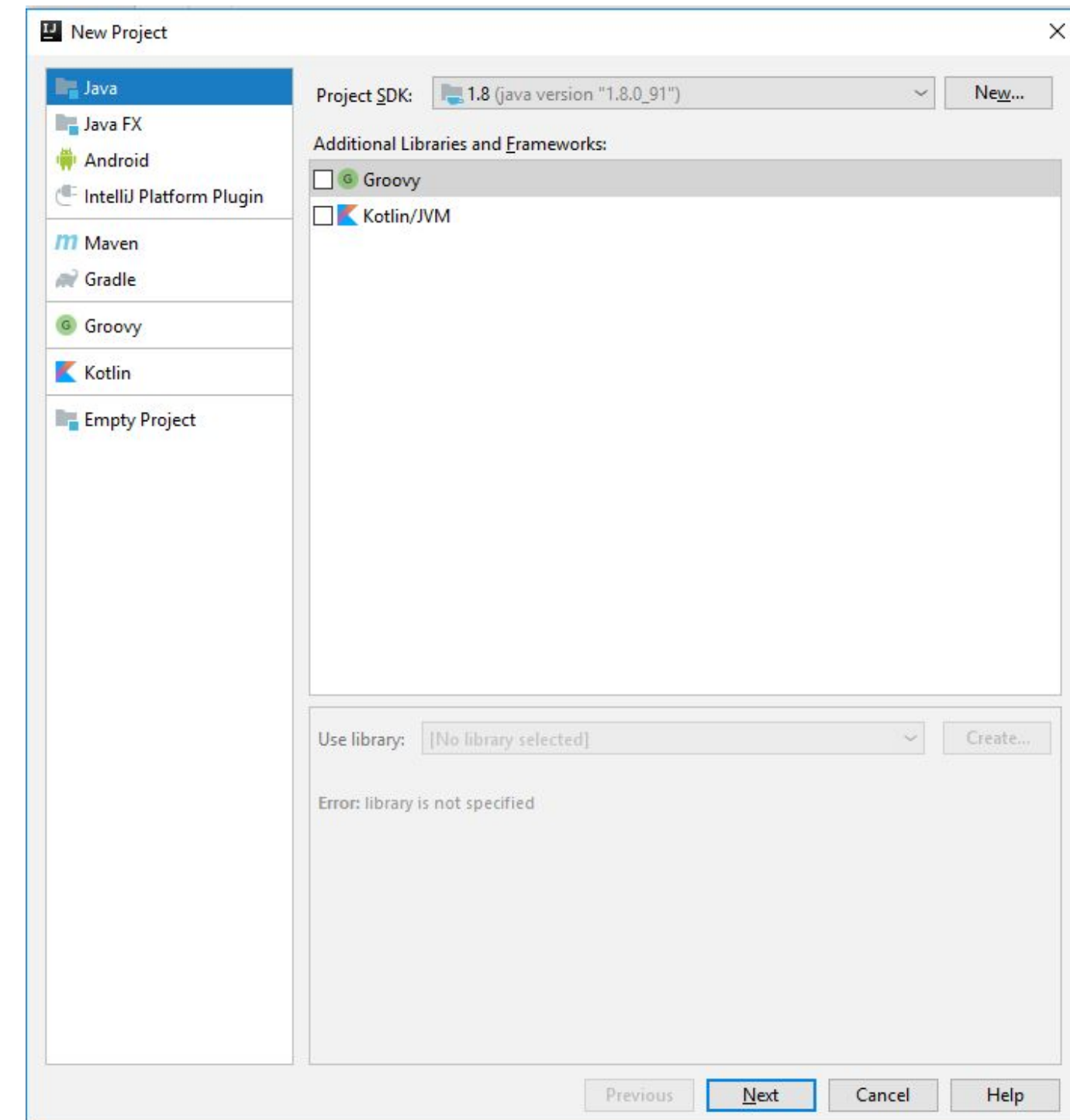
# Maven

- Download and install Maven
- Set M2\_HOME as environment variable (see pic).



# IntelliJ IDEA

- Download **Community** version of IntelliJ IDEA.
- Open IntelliJ IDEA and create a new Maven Project.





# Dependencies

---

- Search for TestNG and Selenium dependencies on <https://mvnrepository.com>
- Choose latest version
- Copy provided xml and paste it in pom.xml
- Download chromedriver



# Annotations

---

@BeforeClass

@BeforeMethod

@BeforeTest

@Test

@AfterMethod

@AfterClass

@AfterTest

# Assertions

---

```
assertTrue(logo.isDisplayed());
```

```
assertFalse(logo.isDisplayed());
```

```
assertEquals("Expected text", "Actual text")
```

```
assertNotEquals("Expected text", "Actual text")
```

```
assertNull(value)
```

```
assertNotNull(value)
```

QUESTIONS?