Cześć!

- Nazywam się Ania Janus-Kurach
- LinkedIn
- https://github.com/AnnaJanus

Agenda

Etap 1: Wprowadzenie

- Co to są testy automatyczne?
- Java - charakterystyka

Etap 2: Testy automatyczne

- Podstawy Programowania w Javie
- Selenium + TestNG
- Pierwszy test automatyczny

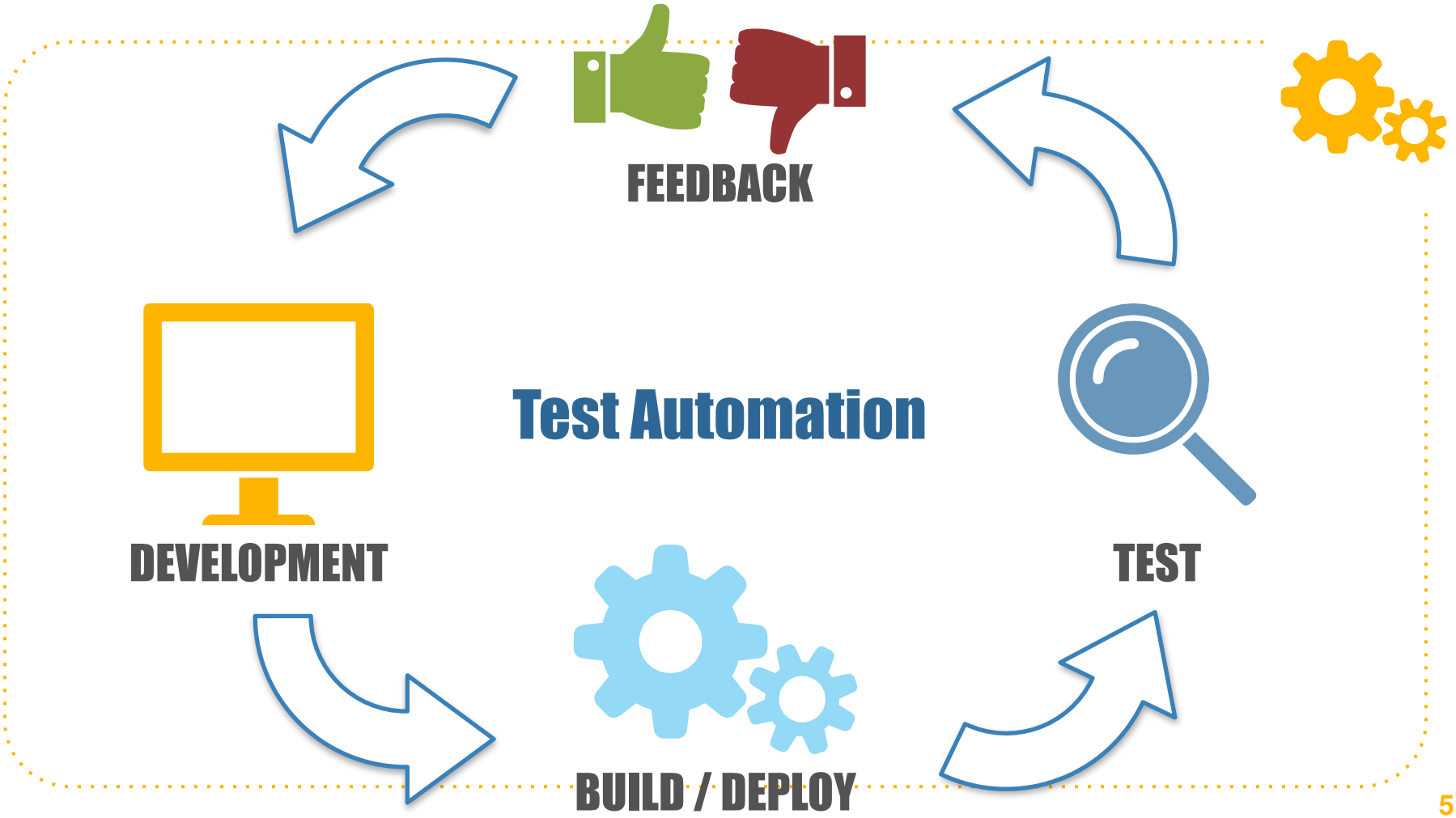
Etap 3: Zakończenie

- Co dalej?
- Tips and tricks

Zasady

- **Szkolenie jest dla Ciebie**
  - pytaj
  - dociekaj
  - próbuj
- **Live Coding**
  - twórz a nie kopiuj
- **Rób notatki**

Co to są testy automatyczne?


FEEDBACK
Test Automation
DEVELOPMENT
TEST
BUILD / DEPLOY

Test Automation

```
@Test
public void oneItemTest(){
    loginSteps.userLogsIntoApplication(properties.getProperty("username"), properties.getProperty("password"));
    productsSteps.userChooseProduct(Product.BackPack);
    headerSteps.userChecksNumberOfItemsInBasket("1");
}
```

Założenia

- **Czytelny, łatwy w zrozumieniu kod**
- **Nie wszystko warto automatyzować**
- **Testy powinny być niezależne**
  aby można było puszczać je równolegle
- **Dane testowe**
  działamy na nowych danych testowych, usuwamy po sobie stare dane testowe
- **Szybki feedback**

Wyzwania

- Lokalizacja elementów strony
- Symulacja akcji użytkownika
- Czekanie
    - na załadowanie strony
    - na efekt wykonania akcji
- Sprawdzenie poprawności wyniku (asercje)

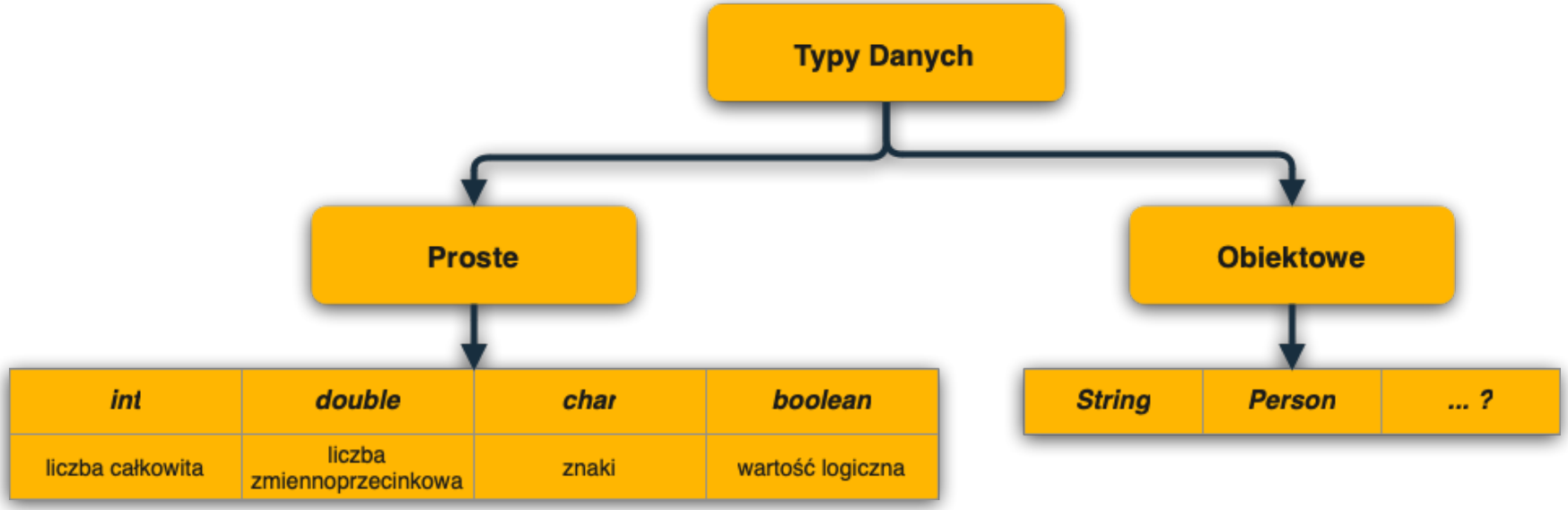
Java - charakterystyka

Java

- **„Napisz raz - uruchamiaj wszędzie” (JVM)**
- Język obiektowy
- Przyjazna składnia
- Automatyczne zarządzanie pamięcią
- Rozwijana przez środowisko deweloperskie (biblioteki)

Java - Podstawy

Typy Danych


Typy Danych
Proste
Obiektowe
int
double
char
boolean
liczba całkowita
liczba zmiennoprzecinkowa
znaki
wartość logiczna
String
Person
... ?

Proste vs Obiektowe

- Pisane z małej litery
- Zawsze przyjmują wartość
- Nie posiadają własnych metod ani właściwości

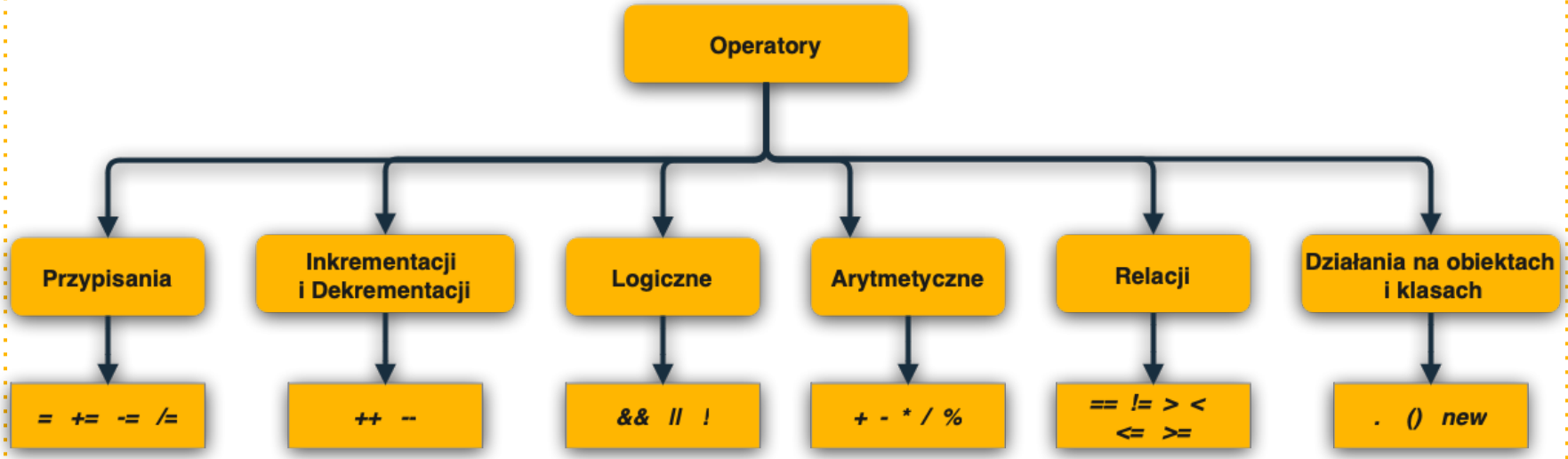
- Pisane z wielkiej litery
- Mogą być **null**
- Posiadają metody i właściwości

Proste vs Obiektowe

- Pisane z małej litery
- Zawsze przyjmują wartość
- Nie posiadają
  metod ani wła

- Pisane z wielkiej litery
- Mogą być **null**
- dają metody i
  iwości

Operatory


Operatory
Przypisania
Inkrementacji i Dekrementacji
Logiczne
Arytmetyczne
Relacji
Działania na obiektach i klasach
= += -= /=
++ --
&& || !
+ - * / %
== != > < <= >=
. () new

Zmienne

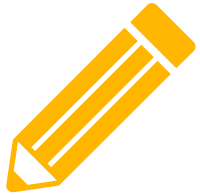
- Rezerwacja miejsca w pamięci
- Deklarujemy, co będziemy przechowywać
- Możemy zmieniać zawartość
- Jedna wartość w jednej zmiennej
- Unikalne nazwy (w obrębie klasy)

Zmienne

- Rezerwacja miejsca w pamięci
- Deklarujemy, co będziemy przechowywać
- Możemy zmieniać zawartość
- Jedna wartość w jednej zmiennej
- Unikalne nazwy (w obrębie klasy)

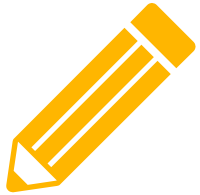
QUIZ

Instrukcje Sterujące



```
if ( zmienna > 5){
System.out.println („Zmienna jest większa od 5”);
}
```

Instrukcje Sterujące

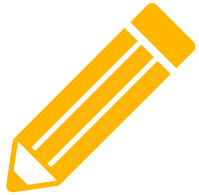


```
if ( zmienna > 5){
    System.out.println („Zmienna jest większa od 5”);
}
```

( ) - w środku znajduje się warunek

{} - w środku znajduje się kod bloku, który ma się wykonać jeśli warunek równy jest true

Instrukcje Sterujące



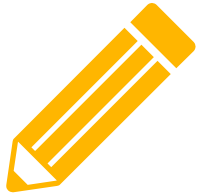
```
if ( zmienna > 5){
    System.out.println („Zmienna jest większa od 5”);
} else {
    System.out.println („Zmienna jest mniejsza od 5”);
}
```

Instrukcje Sterujące - zadanie 1

Co zostanie wyświetlone na ekranie?

```
String zmienna_tekstowa = "Lukasz";
if((zmienna_tekstowa.length())>4){
    System.out.println("Wow");
}else{
    System.out.println("Lol");
}
```

Instrukcje Sterujące - zadanie 2



A. Utwórz zmienną o nazwie ‚number'.

Napisz instrukcję wyświetlającą wartość bezwzględną zmiennej number.

```
|10| = 10
|-10| = 10
Warunek w nawiasach - czy liczba jest dodatnia
if ( zmienna > 5){
    System.out.println („Zmienna jest większa od 5”);
} else {
    System.out.println („Zmienna jest mniejsza od 5”);
}
```

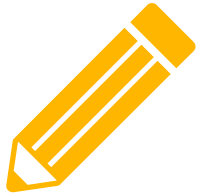
Instrukcje Sterujące

Przykłady użycia w testach automatycznych:

```
if(!checkbox.isSelected()){
    checkbox.click();
}
```

```
if(!testResult.isSuccess()){
    takeScreenshot();
}
```

Metody



- Czynności, które zapisujemy w danej klasie:
  - Klasa Przeglądarka:
    - Zamknij();
    - OtwórzStronę();
    - PobierzLiczbęOtwartychOkien();
  - Klasa Pole Tekstowe:
    - ?

Metody

- Zebrane linijki kodu, które wykonujemy używając nazwy metody

```
public void talk() {
    System.out.println("Name: ");
    System.out.println("Job: ");
    System.out.println("Already In IT: ");
    System.out.println("Programming Languages: ");
    System.out.println("Why this course: ");
}
```

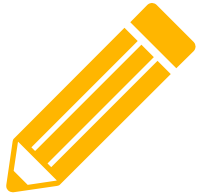
Metody

Widoczność typ_zwracany nazwa() {

kod}

```
public void talk() {
    System.out.println("Name: ");
    System.out.println("Job: ");
    System.out.println("Already In IT: ");
    System.out.println("Programming Languages: ");
    System.out.println("Why this course: ");
}
```

Metody - typ zwracany



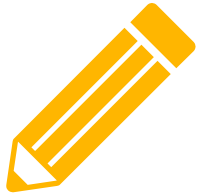
void - metoda nie ma za zadanie zwrócić wartości:

```
public void talk() {
    System.out.println("Name: ");
    System.out.println("Job: ");
    System.out.println("Already In IT: ");
    System.out.println("Programming Languages: ");
    System.out.println("Why this course: ");
}
```

int, String, char, double, Person, Driver - metoda ma za zadanie wykonać operacje i zwrócić ich wynik:

```
public String getTextFromLabel(){
    return label.getText();
}
```

Metody - typ zwracany



void - metoda nie ma za zadanie zwrócić wartości:

- **?**

int, String, char, double, Person, Driver - metoda ma za zadanie wykonać operacje i zwrócić ich wynik:

- **?**

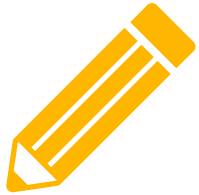
Metody - parametry

Widoczność typ_zwracany nazwa(typ_parametru nazwa_parametru) {

kod}

```
public int countAbsoluteValue(int number){
    if(number > 0){
        return number;
    }else{
        return -number;
    }
}
```

Metody - zadania



A. W klasie Person utwórz metodę, która wyświetla na ekranie przywitanie użytkownika

Typ_zwracany: void

B. W klasie Person utwórz metodę, która zwraca informację, czy użytkownik jest pełnoletni

Typ_zwracany: boolean

C. W klasie Person utwórz metodę, która zwraca informację, czy osoba jest programistą. Warunkiem jest conajmniej roczna praca w IT i znajomość przynajmniej jednego języka programowania

Typ_zwracany: boolean

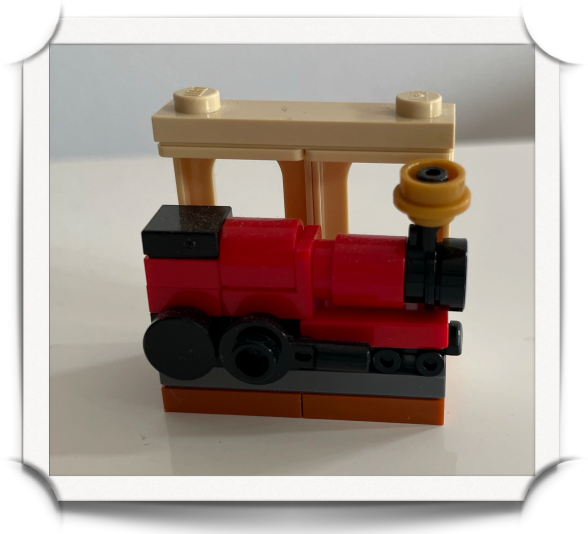
Zadanie

Gdzie jest błąd?

```
public void getPersonalInfo(){
    String info = name + " " + job;
    return info;
}
```

Programowanie Obiektowe

- Każdy element programu jest obiektem
- Obiekty posiadają właściwości i czynności
- Obiekty mogą współpracować ze sobą - wymieniać informacje i wykonywać różne zadania

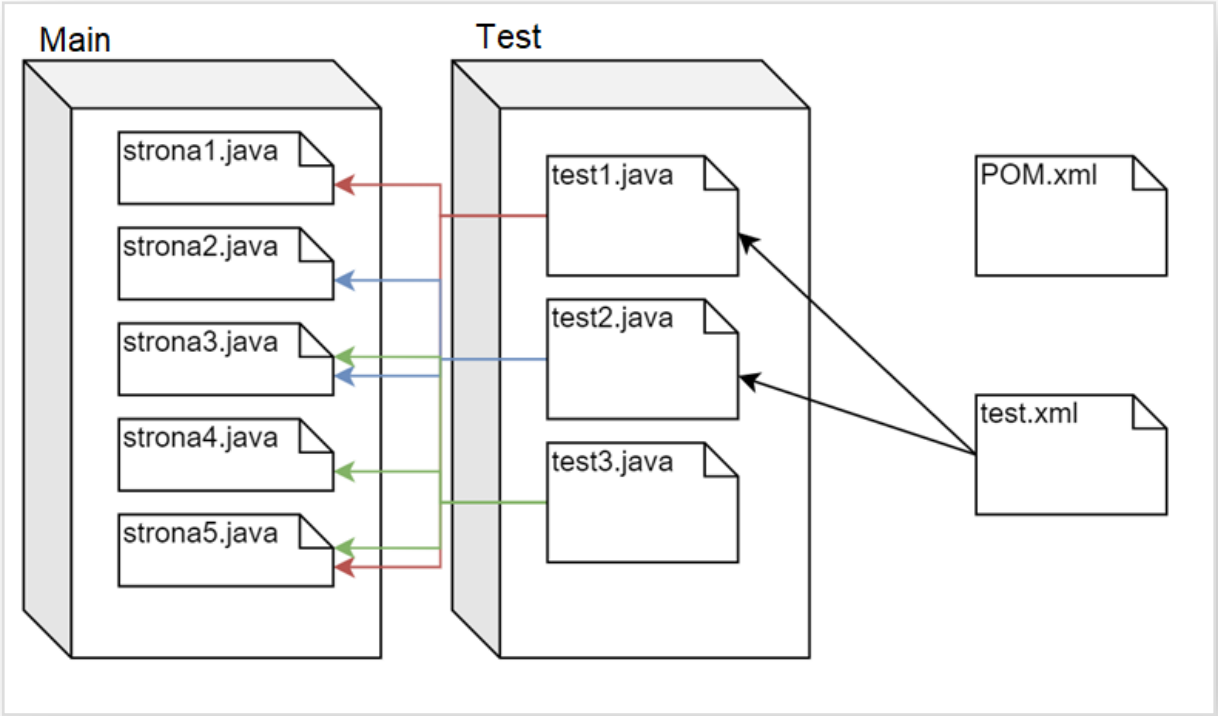


Klasa vs Obiekt

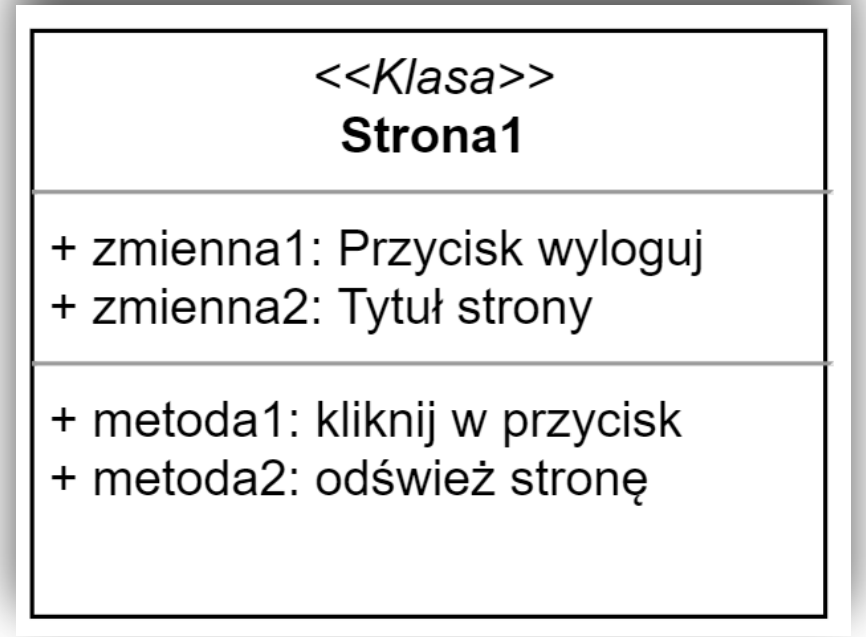
- Klasa - szablon, zawiera właściwości i czynności, które można wykonać na danym obiekcie (np. Person)
- Obiekt - instancja klasy - jednostka reprezentująca element programu (np. Ania)

QUIZ

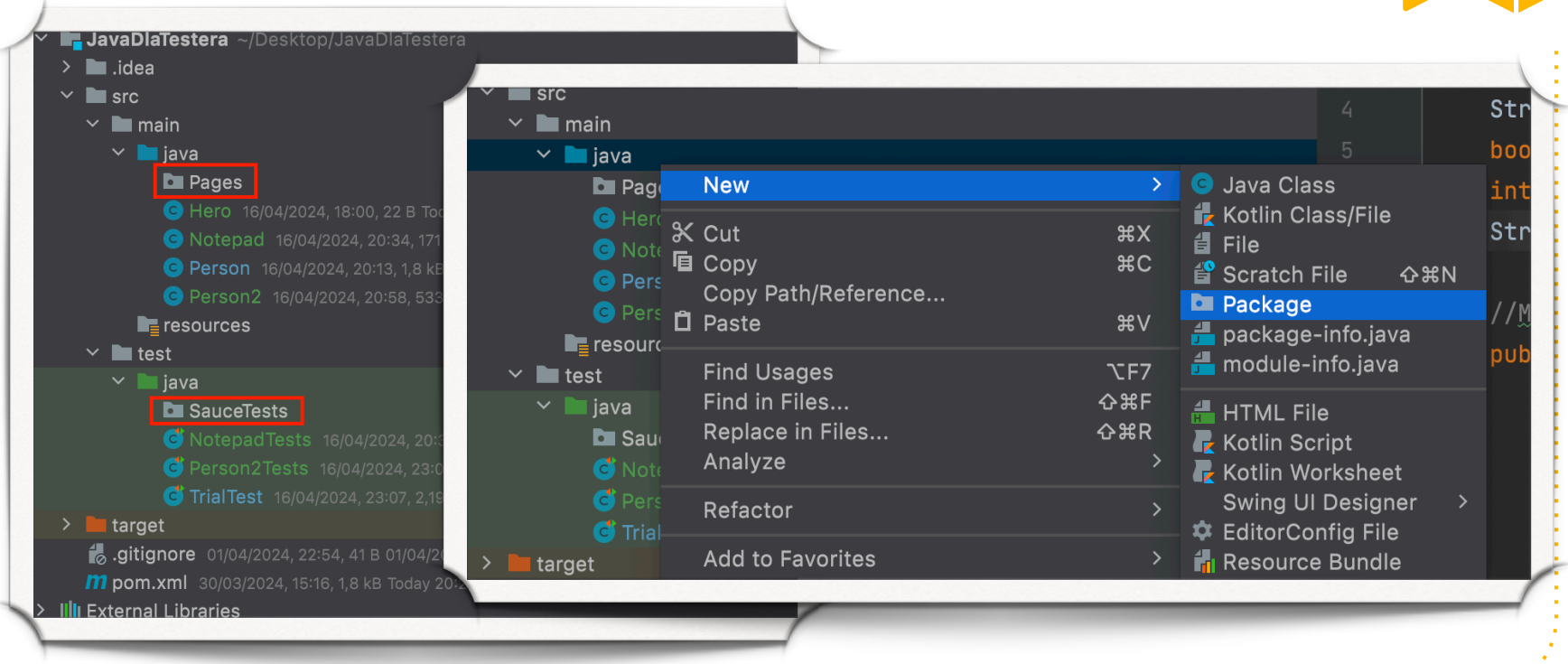
Klasy w testach automatycznych



Klasy w testach automatycznych



Framework - przygotowanie pakietów



Framework - przygotowanie klas

Zadanie: Zautomatyzuj test logowania do aplikacji

Strona: https://www.saucedemo.com/

Step	Expected Result
Provide standard user's username and password	Data is properly provided
Click ‚Login' button	Swag Labs' logo is displayed

Biblioteki

Selenium

Zawiera klasy potrzebne do obsługi przeglądarki

- WebElement
- WebDriver (ChromeDriver, FirefoxDriver, etc..)
- WebDriverWait

```
@Test
public void browserTest() throws InterruptedException {
    ChromeDriver driver = new ChromeDriver();
    driver.get("https://github.com/AnnaJanus/JavaDlaTestera");
    Thread.sleep( millis: 10000);
    driver.close();
}
```

Zadanie

Uruchom przeglądarkę w zmaksymalizowanym oknie

Tip:

Sprawdź jakie funkcje masz dostępne w obiekcie Window:

```
driver.manage().window()…
```

Tryb Headless

- Tryb bez podglądu
- Testy wykonują się szybciej
- Niezbędny jeśli wykonujemy testy na zewnętrznym serwerze

```
ChromeOptions options = new ChromeOptions();
options.addArguments("--headless");
WebDriver driver = new ChromeDriver(options);
```

WebElement

- Obiekt, który reprezentuje element strony
  - `WebElement element;`
- Można na nim wykonywać akcje
  - `element.click()`
  - `element.sendKeys()`
  - `element.clear()`
- Można z niego pobierać dane
  - `element.getAttribute(String attrName)`
  - `element.isEnabled()`
  - `element.getText()`

Framework - implementacja

Zadanie: Zautomatyzuj test logowania do aplikacji

Strona: https://www.saucedemo.com/

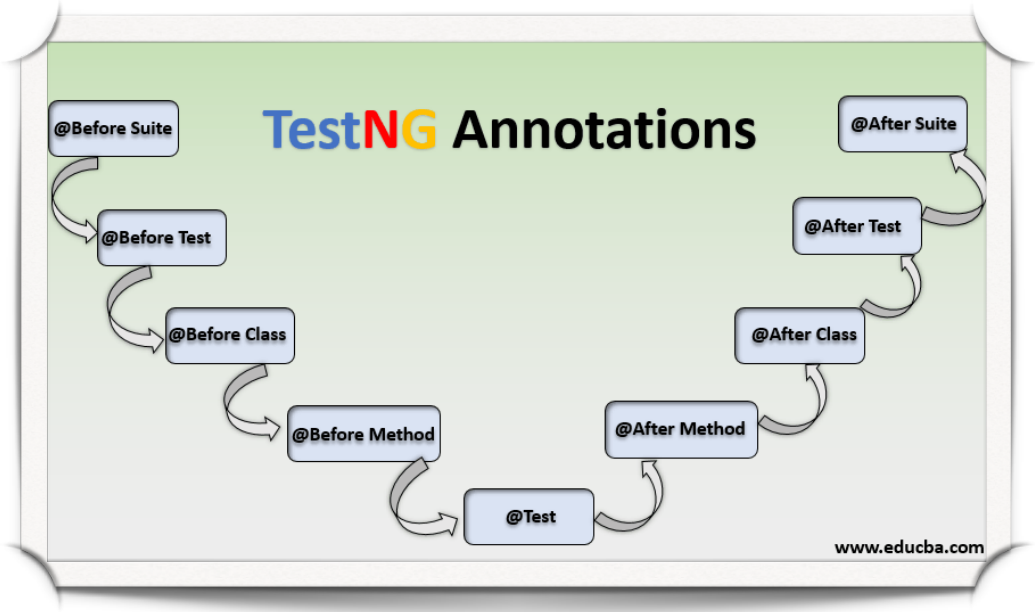
Step	Expected Result
Provide standard user's username and password	Data is properly provided
Click ‚Login' button	Swag Labs' logo is displayed

Element html

```
<a id="login" class="button" href="/user" ng-app="app">Login</a>
```

TestNG

Pozwala na uporządkowanie i uruchamianie testów



@BeforeMethod i @AfterMethod

uruchomi się przed każdą metodą testową

```java
@BeforeMethod
public void setup(){
    browser.get(URL);
}

@AfterMethod
public void cleanUp(){
    browser.close();
}
```

uruchomi się po każdej metodzie testowej
(nawet jak test nie przejdzie)

TestNG

Pozwala na sprawdzenie wyniku testu

```
//                wartość rzeczywista, wartość oczekiwana
Assert.assertEquals(label.getText(),  expected: "Zaloguj");

//                 wyrażenie logiczne
Assert.assertTrue(label.isDisplayed());
```

Framework - Setup, TearDown, Assertion

Zadanie: Zautomatyzuj test logowania do aplikacji

Strona: https://www.saucedemo.com/

Step	Expected Result
Provide standard user's username and password	Data is properly provided
Click ‚Login' button	Swag Labs' logo is displayed

Framework - Move to Page classes

Zadanie: Zautomatyzuj test logowania do aplikacji

Strona: https://www.saucedemo.com/

Step	Expected Result
Provide standard user's username and password	Data is properly provided
Click ‚Login' button	Swag Labs' logo is displayed

Zadanie

Dodaj nowy test, w którym dodajesz jeden z produktów i sprawdzasz czy ikona przy koszyku ma text = 1

Następne Kroki

Co dalej?

1. Rozwijać się jako programista:
    - https://www.codewars.com/
2. Strony do ćwiczenia Selenium:
    - http://webdriveruniversity.com/index.html
    - http://the-internet.herokuapp.com
3. Poznawać nowe frameworki
    - Selenide
    - Playwright
4. Wzorce Projektowe
    - POM
    - PageFactory
    - Dependency Injection

Czego nauczyłaś/nauczyłeś się na szkoleniu?