Web UI - Test

Część I - podstawy Javy:

1. Dodaj do swojego developerskiego katalogu ‘TechnicznyTester’ nowy katalog ‘test’.
2. W Intellij stwórz nowy Maven project wewnątrz katalogu ‘test’.
3. Wewnątrz katalogu ‘main/java’ stwórz nowy pakiet np. ‘Dev’ i dodaj nową klasę ‘HelloWorld’.
4. W klasie ‘HelloWorld’ dodaj nową metodę statyczną main, która będzie wyrzucała na ekran tradycyjne powitanie ‘HelloWorld’. Odpalamy klasę klikając prawym przyciskiem na okienko z klasą i wybieramy ‘Run’
   1. Co to oznacza, że metoda jest statyczna?
   2. Co zwróci metoda typu void?
   3. Jakie znasz inne typy zwracanych wartości?
5. Zadeklaruj dwie zmienne publiczne, statyczne typu int np. x i y i przypisz do nich wartości liczbowe. Wypisz na ekran ich sumę.
   1. Co się stanie gdy zmienne nie będą statyczne?
   2. Co się stanie gdy zmienne będą miały modyfikator dostępu private
   3. Jakie znasz inne modyfikatory dostępu?
6. Wykonaj to samo co powyżej ale poprzez użycie osobnej metody ‘sum’. Metoda ta ma zwracać wartość typu int i być wywoływana przez metodę main.
   1. Zmodyfikuj wywołanie metody sum w metodzie main tak by bezpośrednio podawać wartości do sumy z pominięciem deklaracji
   2. Zmodyfikuj wartości zmiennych x i y tak by nie były typu int i zwracały liczbę niecałkowitą
   3. Co by się stało gdybyśmy zmienili wartości zmiennych x i y na niecałkowite ale jednocześnie pozostawili zmienne x i y typu int?
7. Napisz kolejną metodę ‘counter’, która wywyoływana z metody main będzie wypisywała w pętli liczby od 1 do 10
   1. Jak to wykonasz w pętli for?
   2. A jak w pętli while?
8. Zmodyfikuj powyższą metodę tak by wypisywała w wartości przemnożone przez siebie np. 1\*1, 2\*2 itd. Jeśli ten iloczyn będzie większy niż 20 to nie wypisuj na ekran
   1. Użyj instrukcji If Else
   2. Użyj instrukcji Switch
9. Napisz kolejną metodę ‘primeNumbers’, która wypisze na ekran tylko liczby pierwsze z zadeklarowanego zakresu.
10. Przerzuć metodę z liczbami pierwszymi do nowej klasy np. ‘PrimeNumbers’ i w metodzie main klasy ‘HelloWorld’ wywołaj tę metodę. Uwaga będziesz musiał stworzyć nowy obiekt (zmienną obiektową) typu PrimeNumbers
    1. Co się stanie jak metoda ‘primeNumbers’ przestanie być statyczna?
11. A teraz skomplikuj sprawę jeszcze bardziej. Stwórz kolejną klasę np. ‘PrimeNumbersPrimordial’, z której to klasy klasa ‘PrimeNumbers’ będzie dziedziczyła metodę ‘primeNumbers’.
    1. Wywołaj metodę ‘primeNumbers’ w metodzie main
    2. Co się wydarzy gdy zmienimy modyfikator dostępu do klasy ‘PrimeNumbersPrimordial’ na private
    3. Wywołaj zmodyfikowaną metodę ‘primeNumbers’ w metodzie main. Zawartość metody ‘primeNumbers’ powinna się zmienić w klasie ‘PrimeNumbers’
12. Dodaj do klasy ‘PrimeNumbers’ dwie nowe metody o tej samej nazwie np. ‘loadMethod’ ale o rożnej ilości parametrów. Pierwsza przyjmuje tylko jeden typu int x, druga natomiast dwa int x i int y.
    1. Pierwsza metoda wywołana w metodzie main będzie zwracała jedynie wartość przemnożoną przez siebie. Natomiast druga będzie zwracała kwadrat sumy x i y.
    2. Dodaj do pierwszej metody pętlę od 1 do 5 i pomnóż każdą kolejną wartość iteracji o wartość podaną w argumencie metody. Jeśli wynikiem takiego działanie będzie liczba większa od 10 przerwij wykonywanie pętli
    3. Zmodyfikuj tę metodę używając zamiast warunku if else, konstrukcji try catch
13. A teraz zadanie na piątkę - stwórz metodę z listą użytkowników (nazwiska) ‘myList’ w nowej klasie ‘MyPersonalList’. Wywołaj następnie w metodzie main metodę ‘myList’ z następującym założeniem, że ma metoda zwrócić jedynie takie nazwiska, które zawierają frazę ‘Mar’

Część II - podstawy Selenium i PageObject Pattern:

1. Stwórz w katalogu test/java nowy pakiet, który nazwiemy ‘test’
2. Następnie stwórz w tym pakiecie strukturę katalogów TechnicznyTester/pages i TechnicznyTester/tests
3. Do katalogu src/test dodajmy jeszcze jeden katalog ‘resources’
4. Będziemy musieli na wstępie zmodyfikować nieco nasz plik pom.xml, w którym zawarte są pakiety i biblioteki do zaciągnięcia. Będziemy potrzebowali: selenium-java, selenium-api, selenium-server, selenium-support wszystko w wersji 3.3.0, dalej testng w wersji 6.8.8.
5. Następnie zaciągnij ChromeDriver w wersji kompatybilnej z wersją selenium i testng i wrzuć go do katalogu ‘resources’
6. Mamy przygotowaną projekt do kodowania