Zadanie 1 - DriverFactory

Stwórz **enuma** Browser i dodaj w nim *CHROME, FIREFOX, IE, EDGE*.

Zaimplementuj klasę DriverFactory i w niej metodę getDriver(Browser browser). Ona w zależności od przekazanego enuma powinna stworzyć innego drivera i go zwrócić.

DriverFactory powinno być wywoływane w TestBase ale nie hardcodujemy w niej konkretnego enuma ale powinniśmy pobierać jego wartość z pliku YML, o który prosił was Darek.

Czyli jak w YML mamy browser=chrome to wtedy parsujemy to na enuma i trafia on do DriverFactory :)

P.S. zaprojektujcie tak klasę DriverFactory, żeby nie miała ona 1 potężnej metody getDriver tylko dodatkowo klika mniejszych prywatnych, przemyślcie jak to ładniej zrobić :)

Zadanie 2 - UserBuilder

Przeczytać jak implementuje się wzorzec projektowy Builder.

W pakiecie models stworzyć klasę User (z polami które są wymagane do testu rejestracji). Sworzyć klasę UserBuilder, która będzie tworzyć obiekty klasy User.

Następnie stworzyć klasę UserFactory, a w niej metody gerRandomUser() oraz getAlreadyRegisteredUser(). Obydwie metody będą zwracać obiekt klasy User, a do jego tworzenia będą używać UserBuildera.

Metoda getRandomUser() powinna zwracać losowo wygenerowanego usera, losujemy dane za pomocą biblioteki: Java Faker.

Metoda getAlreadyRegisteredUser() powinna zwrócić użytkownika, który już ma konto (możecie jego dane zahardcodować)

Przydatne linki:

https://devcave.pl/effective-java/wzorzec-projektowy-builder

https://www.baeldung.com/java-faker

Search tests

6. **Search test**
   - random one name of product from list of products on home page
   - enter product name into search field
   - click search button
   - check in the search result there is name of product that you are searching for
7. **Search test - dropdown**
   - random one name of product from list of products on home page
   - enter product name into search field
   - check in dropdown results there is name of product that you are searching fo

Product and categories tests

8. **Categories**
    - iterate through each category (with loop)
    - each time check if name of opened category matches what you clicked
    - each time check if filters side menu is displayed
    - each time check count how many products is in category and check if label “There are X products” contains correct X number of products in that category
    - now do the same for subcategories (under clothes and accessoriess)
9. **Filters**
    - go to Art category
    - Select $8.00 - $10.00 filter
    - check if now in list of products there are only products that matched selected price filter
    - clear filter
    - repeat that for remaining price filters
10. **Prices drop**
    - click on ‘prices drop’ on the footer
    - check if “on sale” pages loaded
    - check if “-20%” is displayed on each product image
    - check if each product has regular and discounted price displayed
    - for each product calculate if actual price is 20% lower than regular
    - open one of discounted products
    - check if product page has “SAVE 20%” label
    - check if product page has regular and discounted price displayed
    - calculate if actual price is 20% lower than regular

Basket and checkout tests

Design class that will store what is already added to cart. It need to keep list of products with names, prices, quantity and total order cost. Please remember that if you will be randomizing products to add to basket sometimes can happen that you will random the same product. In that situation when you will update your object that stores date from basket you cannot add new element to list. But you need to add quantity of product that is already on the list.

22. **Product successfully added to your shopping cart**
    - open random product from random category
    - set quantity with random value from 1 to 5
    - click add to card button
    - check if popup has correct: name, price, quantity, there are X items in your cart, Total products value
    - click continue shopping
    - check if cart icon has update quantity of products – Cart (X)
    - repeat previous steps (random -> add -> check popup -> close -> check cart icon) 3 times (so u will have up to 15 products in cart

23. **Basket**
    - add 5 random products to basket (remember all details)
    - click on basket icon to open basket page
    - check if list of items on basket page have all products that you have added and products have all correct details (names, prices quantities)
    - check if total order value is correct
    - set quantity of first product to 5
    - check if total order value is correct
    - check if clicking on arrow that adds one product is updating quantity value
    - check if total order value is correct
    - check if clicking on arrow that removes one product is updating quantity value
    - check if total order value is correct
    - start deleting each product one by one using trash icon and after deleting each product check if total order price is calculated correctly
24. **Checkout**
    - register new user
    - add 5 random products with random quantity from 1 to 3
    - on adding last product click on proceed to checkout button on product popup
    - fill address form
    - select shipping method
    - select Pay by bank wire
    - click on Terms and conditions of use
    - check in popup text is not empty
    - close popup
    - accept terms of use
    - click order button
    - on order confirmation page check if all added products are displayed and have all correct details (names, prices quantities)
    - check if Payment method and Shipping method is matching what you selected in last step
    - save Order reference number
    - go to order history
    - find Order reference on the list
    - check if date, total price, payment and status is correct
    - click on details
    - check if all added products are displayed and have all correct details (names, prices quantities)
    - check if Delivery address and Invoice address have values that you used in checkout