При написании автотестов использовать паттерн Page Object.  
  
Тест 1:  
1)Перейти на сайт 21vek.by.  
2)Раскрыть список “Компьютеры”.  
3)Перейти Компьютерная техника -> Ноутбуки.  
4)Выставить диапазон цен от 1200 – 6840.

- на данном шаге можно было бы провести проверку, что минимальный диапазон должен быть меньше или равен максимальному;

- также оба числа должны быть положительны;

Однако, данная проверка заложена на уровне сайта.  
5)Установить чекбокс “В наличии”.  
6)Установить чекбокс Производители -> Lenovo.  
7)Раскрыть список “Линейка”, выбрать чекбоксы IdeaPad L (Lenovo),ThinkPad E (Lenovo), ThinkPad X (Lenovo).

- данный шаг пришлось дополнить нажатием кнопки «Показать всё»  
8)Раскрыть список “Тип”, выбрать чекбокс “ультрабук”.

9)Кликнуть кнопку “Показать товары (кол-во товаров)”.

10)Кликнуть на кнопку “В корзину” на [Ноутбук Lenovo ThinkPad X1 Carbon Gen 8 (20U90006RT)](https://www.21vek.by/notebooks/thinkpadx1carbongen8_lenovo_04.html).  
11)Перейти в корзину проверить в корзине наличие [Ноутбук Lenovo ThinkPad X1 Carbon Gen 8 (20U90006RT)](https://www.21vek.by/notebooks/thinkpadx1carbongen8_lenovo_04.html).

- перед данным шагом производится обновление страницы, т.к. если зайти в корзину сразу, то товар может быть не найден и корзина окажется пустой.  
12)Сверить стоимость товара.

- сверка товаров происходит при получении значения стоимости на страницы «ноутбуки», и при переходе на страницу корзины;

- используя метод *GetStringWithoutLetters* класса *StringUtils*, из полученных значений удаляются лишние символы (пробелы и буквы).  
13)Кликнуть кнопку “Оформить заказ”.  
14)Кликнуть кнопку “Подтвердить заказ ->”.  
15)Проверить наличие сообщений о незаполненных обязательных полях.

- на данном шаге производится проверка, есть ли хоть одно незаполненное поле, если данное поле имеется (в тесте 4 незаполненных поля), то тест проходит проверку.  
  
Тест 2:  
1)Перейти на сайт 21vek.by.  
2)Ввести в поле Поиск слово “National”.

- ввиду того что при вводе данного слова не происходит отображения фенов данной фирмы, наименование было заменено на *"Фен Philips HP"*.  
3)Кликнуть Search (Submit) button.   
4)Кликнуть под товаром [Профессиональный фен National NB-HD2202](https://www.21vek.by/hairdryers_brushes/nbhd2202_national.html) по ссылке “Узнать о поступлении”.

- полное наименование фена было заменено на *"Фен Philips HP8233/00"* (который на момент написания тестов был помечен как «нет на складе»)  
5)Кликнуть кнопку “Отправить”.  
6) Проверить наличие сообщений о незаполненных обязательных полях.

- на данном шаге производится проверка, есть ли хоть одно незаполненное поле, если данное поле имеется (в тесте 2 незаполненных поля), то тест проходит проверку.  
7)Ввести в поле “Имя” и “E-mail” корректные значения.  
8)Кликнуть кнопку “Отправить”.  
9)Проверить наличие в модальном окне “Узнать о поступлении” текста:  
Если товар появится на складе, вам придет сообщение на почту.

- на данном шаге указывается email в тесте [email21vek@mail.com](mailto:email21vek@mail.com), если на товар нажать «узнать о поступлении», то должен появится текст «Если товар появится на складе, вам придет сообщение на почту», если же сделать это вторично, то «Вы уже подписаны на обновления информации об этом товаре», для этого нужно каждый раз изменять почту, либо вручную, либо добавить метод в StringUtils для генерации нового почтового адреса.

- в данном случае тест упадет, т.к. данный юзер уже нажимал кнопку «узнать о поступлении».  
10)Кликнуть кнопку “Закрыть”.  
11)Проверить под товаром [Профессиональный фен National NB-HD2202](https://www.21vek.by/hairdryers_brushes/nbhd2202_national.html)  наличие текста “В листе ожидания”.

Дополнительно:

1. В данном фреймворке был использован паттерн PageObject, что позволило отделить логику страницы от тестов.
2. В данном фреймворке был использован паттерн PageElement, что позволило дробить такие части как Header, Nav, ModalWindow и использовать в главной родительской странице Page, а следовательно и взаимодействовать с данными повторяющимися частями страницы.
3. Если потребуется упростить логику фреймворка для понимания, а также ускорить загрузку некоторых элементов через кэширования элементов в память, можно использовать паттерн PageFactory.

Links: https://www.lambdatest.com/blog/page-object-model-tutorial-selenium-csharp/

<https://www.toolsqa.com/selenium-webdriver/c-sharp/pagefactory-in-c/>

1. В данном фреймворке не была реализована параллельность тестов.
2. Драйвер для браузеров Edge не всегда работает корректно.
3. Скорость работы тестов в Google Chrome быстрее чем в остальных браузерах.
4. В процессе написания фреймворка были использованы паттерны: PageObject, PageElement, Singleton, Factory, Decorator.
5. Запуск тестов происходит при переходе во вкладку Test -> TestExploer -> Run All Test in View
6. Если тест не был пройден, то сохраняется скриншот по пути: \Resources\Screenshots\