Классы эквивалентности для регистрации имени пользователя на google.com

По допустимым символам:

1. Комбинации букв, цифр и точек
2. Комбинации букв, цифр, точек и специальных символов

По количеству символов:

1. Менее 6 символов
2. Более 30 символов
3. От 6 до 30 символов

Граничные значения

1)6 символов

2)30 символов

Необходимо провести тестирование на следующих данных

Для логина:

1. Комбинация точек, букв и цифр, от 6 до 30 символов
2. Комбинации букв, цифр, точек и специальных символов, от 6 до 30 символов
3. Комбинация точек, букв и цифр, меньше 6 символов
4. Комбинация точек, букв и цифр, от 6 до 30 символов
5. Комбинация точек, букв и цифр, больше 30 символов
6. Комбинация точек, букв и цифр, 30 символов
7. Комбинация точек, букв и цифр, 6 символов

Классы эквивалентности для регистрации пароля пользователя на google.com

По количеству символов:

1. Менее 6 символов
2. Более 100 символов
3. От 6 до 100 символов

Граничные значения

1) 6 символов

2) 100 символов

Необходимо провести тестирование на следующих данных

Для пароля:

1. Пароль длинной менее 6 символов
2. Пароль длинной более 100 символов
3. Пароль длинной от 6 до 100 символов
4. Пароль длинной 6 символов
5. Пароль длинной 100 символов

Сценарии тестирования для [www.onlinetrade.ru](http://www.onlinetrade.ru).

|  |  |
| --- | --- |
| Идентификатор | Регистрация на сайте www.onlinetrade.ru |
| Краткое описание | Тестирование регистрации на сайте www.onlinetrade.ru с корректным заполнением всех полей |
| Предусловие | Сайт www.onlinetrade.ru открыт в браузере |
| Шаги | 1. Кликнуть на кнопку "Вход в личный кабинет" |
| 2. Кликнуть на кнопку "Регистрация" |
| 3. В поле "Ваше имя" ввести имя "Иван" |
| 4. В поле "E-mail" ввести "testmail@mail.ru" |
| 5. В поле "Пароль" ввести "testPassword1" |
| 6. В поле "Мобильный телефон" ввести "84238934244" |
| 7. Кликнуть на кнопку "Зарегистрироваться" |
| Ожидаемый результат | Успешная регистрация |

|  |  |
| --- | --- |
| Идентификатор | Вход на сайт www.onlinetrade.ru |
| Краткое описание | Тестирование авторизации на сайте www.onlinetrade.ru под существующим аккаунтом |
| Предусловие | 1. На сайте www.onlinetrade.ru зарегистрирован аккаунт. 2. Сайт www.onlinetrade.ru открыт в браузере |
| Шаги | 1. Кликнуть на кнопку "Вход в личный кабинет" |
| 2. В поле "E-mail" ввести "testmail@mail.ru" |
| 3. В поле "Пароль" ввести "testPassword1" |
| 4. Кликнуть на кнопку "Зарегистрироваться" |
| Ожидаемый результат | Успешная авторизация |

|  |  |
| --- | --- |
| Идентификатор | Добавление адреса доставки |
| Краткое описание | Тестирование функции добавления адреса доставки |
| Предусловие | 1. Сайт открыт в браузере. 2. Аккаунт авторизован.  3. Открыт личный кабинет |
| Шаги | 1. Перейти во вкладку "Адреса доставки" |
| 2. Кликнуть по кнопке "Добавить адрес" |
| 3. В поле "Название доставки" ввести "Тест" |
| 4. В поле "Контактный телефон" ввести "89142276584" |
| 5. Кликнуть по кнопке "Сохранить" |
| Ожидаемый результат | Адрес сохранен |

Чек-лист.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Проверка | Chrome Desktop | Yandex Browser Desktop | Opera Desktop | Chrome Mobile |
| Регистрация | Успешно | Успешно | Успешно | Успешно |
| Авторизация | Успешно | Успешно | Успешно | Успешно |
| Добавление адреса доставки | Успешно | Успешно | Успешно | Успешно |

Баг репорт.

|  |  |
| --- | --- |
| Заголовок | При выходе со страницы задания открытый документ не закрывается. |
| Проект | Google class |
| Компонент приложения | Приложение |
| Номер версии | 8.0.341.20.90.4 |
| Критичность | Не критический |
| Приоритет | низкий |
| Статус | Новый (баг-репорт только добавлен) |
| Автор | Лунёв В.В. |
| Назначен на | Иванов В.В. |
| Как воспроизвести | 1. Открыть назначенное задание 2. Загрузить файл.  3. Открыть файл.  4. Нажать кнопку "Back" браузера. |
| Открылась предыдущая страница, поверх нее открытый файл. |
| Открылась предыдущая страница, открытый файл закрылся. |
| Дополнительные сведения | OS: Windows 10 Browser: Chrome 106.0.5249.119 |
| Прикрепленный файл | bug.png |

Листинг функций для тестирования

def fib(n):  
 arr = [1, 1]  
 if(n <= 2 and n >=0):  
 return arr[0:n:];  
 elif(n > 2):  
 for i in range(n - 2):  
 arr.append(arr[len(arr) - 1] + arr[len(arr) - 2])  
 return arr  
 else:  
 raise ValueError

def puzirek(arr):  
 if(type(arr) != list):  
 raise ValueError  
 for i in range(len(arr) - 1):  
 for j in range(len(arr) - i - 1):  
 if arr[j] > arr[j + 1]:  
 arr[j], arr[j + 1] = arr[j + 1], arr[j]  
 return arr

def calc (a, b, sign):  
 print(type(b))  
 if((type(a) != int and type(a) != float) or (type(b) != int and type(b) != float)):  
 raise ValueError  
 if(sign == '+'):  
 return a + b  
 elif(sign == '-'):  
 return a - b  
 elif(sign == '\*'):  
 return a \* b  
 elif(sign == '/'):  
 return a/b  
 else:  
 raise ValueError

Листинг тестовых классов

import pytest  
from labwithtests.lib import fib as f  
  
  
#Тест функции, возвращающей список из n первых чисел Фибоначчи  
class TestFib:  
 #Тест функции на корректных данных (n > 0)  
 def test\_on\_correct\_n(self):  
 assert f.fib(5) == [1, 1, 2, 3, 5]  
 #Тест функции на граничных данных (n == 0)  
 def test\_on\_border\_n(self):  
 assert f.fib(0) == []  
 #Тест функции на не корректных данных (n < 0), функция вызывает ValueError  
 def test\_on\_incorrect\_n(self):  
 with pytest.raises(ValueError):  
 f.fib(-10)

import pytest  
from labwithtests.lib import puzirek as p  
  
  
#Тест функции, реализующей сортировку пузырьком  
class TestPuzirek:  
 # Тест функции на корректных данных (arr - список)  
 def test\_on\_correct\_arr(self):  
 assert p.puzirek([1,5,2,3,6,3,4]) == [1,2,3,3,4,5,6]  
 # Тест функции на не корректных данных (arr - не список)  
 def test\_on\_incorrect\_arr(self):  
 with pytest.raises(ValueError):  
 p.puzirek(25)

import pytest  
from labwithtests.lib import calc as c  
  
  
#Тест калькулятора  
#Корректные данные для a и b - любые числа (int, float)  
#Корректные данные для sign - '+', '-', '\*', '/'  
class TestCalc:  
 #Тест сложения при корректных данных  
 def test\_plus\_on\_correct\_a\_and\_b(self):  
 assert c.calc(5, 4,'+') == 9  
 #Тест сложения при не корректных данных  
 def test\_plus\_on\_incorrect\_a\_and\_b(self):  
 with pytest.raises(ValueError):  
 c.calc(5, '4','+')  
 #Тест вычитания при корректных данных  
 def test\_minus\_on\_correct\_a\_and\_b(self):  
 assert c.calc(5, 4,'-') == 1  
 #Тест вычитания при не корректных данных  
 def test\_minus\_on\_incorrect\_a\_and\_b(self):  
 with pytest.raises(ValueError):  
 c.calc(5, '4', '-')  
 #Тест умножения при корректных данных  
 def test\_multiply\_on\_correct\_a\_and\_b(self):  
 assert c.calc(5, 4,'\*') == 20  
 #Тест умножения при не корректных данных  
 def test\_multiply\_on\_incorrect\_a\_and\_b(self):  
 with pytest.raises(ValueError):  
 c.calc(5, '4','\*')  
 #Тест деления при корректных данных  
 def test\_divide\_on\_correct\_a\_and\_b(self):  
 assert c.calc(5, 4,'/') == 5/4  
 #Тест деления при не корректных данных  
 def test\_divide\_on\_incorrect\_a\_and\_b(self):  
 with pytest.raises(ValueError):  
 c.calc(5, '4', '/')  
 #Тест не корректного знака  
 def test\_incorrect\_sign\_on\_correct\_a\_and\_b(self):  
 with pytest.raises(ValueError):  
 c.calc(5, 4, 'Б')

Работа тестовых классов

