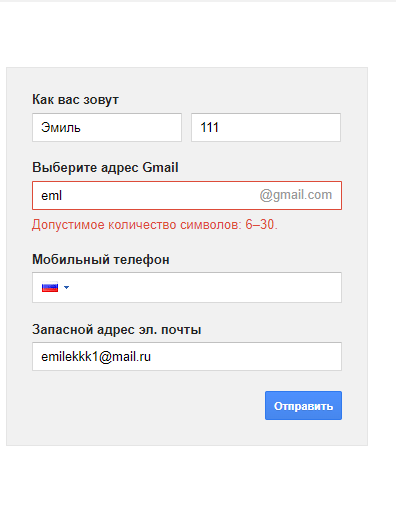
**Задание №1.** Выбрать форму регистрации сайте, например gmail.com или yandex.ru. Найти классы эквивалентности и граничные значения для полей. Определить, на каких данных проводить тестирование этой формы.

Проверим форму регистрации на сайте gmail.ru



Будем тестировать поле ввода адреса почты.

Гранчиные значения для поля имени пользователя:

1. Минимум – 6;
2. Максимум – 30.

Классы эквивалентности для поля имени пользователя:

1. Буквы латинского алфавита;

2. Цифры;

3. Точки.

**Задание №2.** Выбрать ПО и составить 3 сценария тестирования и чек-лист для них.

Переходим на сайт gmail.ru, выбираем поле “Выберите адрес Gmail” и проводим следующие тесты:



**Задание №3.** Выбрать ПО и составить для него баг-репорт.

|  |  |
| --- | --- |
| Заголовок | При попытке открыть картинку в мессенджере на сайте vk.com фотография не показывается, темнеет экран и не появляется кнопка “закрыть фотографию” |
| Проект | vk.com |
| Компонент приложения | Открытие картинки |
| Номер версии | Версия программы, на которой была найдена ошибка |
| Критичность | Блокирующий |
| Приоритет | Высокий |
| Статус | Закрыт (исправлен) |
| Автор | Бадретдинов Э.К. |
| Назначен на | Невоструев И.Д. |
| Как воспроизвести | 1. Войти в веб-версию vk через браузер на ПК.  2. Перейти в диалог с любым человеком.  3. Открыть любую картинку из диалога  Полученный результат:  Картинка открылась  Ожидаемый результат:  Картинка не открылась, экран потемнел, кнопки на сайте неактивны |
| Дополнительные сведения | Информация об окружении, на котором был найден баг: OS Windows 10 Корпоративная 22H2, Yandex Browser 22.11.2.807 (64-bit). |
| Прикрепленный файл |  |

**Задание №4.** Реализовать небольшую библиотеку функций. В библиотеку включить следующие функции:

* определения n чисел Фибоначчи – функция принимает n, возвращает список из чисел
* сортировку пузырьком  функция принимает список чисел, возвращает его отсортированную копию
* калькулятор  функция принимает 3 аргумента: число 1, число 2 и знак действия: +, -, \*, / выполняет действие и возвращает результат

Реализовать unit-тесты данных функций при помощи библиотеки pytest. Реализовать тестирование реакции программы на корректные и не корректные входные данные. Определить классы эквивалентности и граничные значения для входных данных. Код сопроводить комментариями.

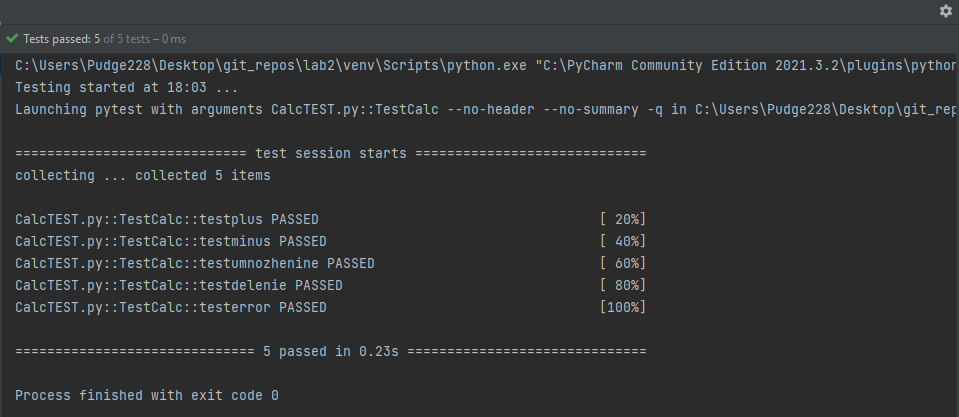
Класс функций:

import numpy as np  
import random  
  
# Ввод чисел в формате a + b  
def calc(a,sign,b):  
 if sign == "+":  
 return (a+b)  
 if sign == "-":  
 return (a-b)  
 if sign == "\*":  
 return (a\*b)  
 if sign == "/":  
 return (a/b)  
  
def bubble(way):  
 a = [5,3,6,7,2,4,1,8]  
 # Сортируем по возрастанию  
 if way == 1:  
 for i in range(len(a) - 1):  
 for j in range(len(a) - i - 1):  
 if a[j] > a[j + 1]:  
 a[j], a[j + 1] = a[j + 1], a[j]  
 # Сортируем по возрастанию  
 elif way == 2:  
 for i in range(len(a) - 1):  
 for j in range(len(a) - i - 1):  
 if a[j] < a[j + 1]:  
 a[j], a[j + 1] = a[j + 1], a[j]  
 else:  
 return -1  
 # Выводим  
  
 return a  
  
def Fibonacci(n):  
 a = np.zeros(n)  
 # Инициализируем первые 3 числа  
 a[0] = 1  
 a[1] = 1  
  
 i = int(2)  
  
 # Считаем числа  
 while i < n:  
 a[i] = a[i - 1] + a[i - 2]  
 i = i + 1  
  
 # Возвращаем массив  
 return a.tolist()

Класс тестов калькулятора:

import Calculate  
import pytest  
#тест калькулятора  
class TestCalc:  
 #задаем числа и знак действия  
 @pytest.fixture()  
 def plus(self):  
 return 5,'+',5  
  
 @pytest.fixture()  
 def minus(self):  
 return 10,'-',10  
  
 @pytest.fixture()  
 def umnozh(self):  
 return 4,'\*',5  
  
 @pytest.fixture()  
 def delen(self):  
 return 10,'/',10  
  
 @pytest.fixture()  
 def error(self):  
 return '+', '-', '+'  
   
 #тест плюса  
 def testplus(self,plus):  
 assert Calculate.calc(\*plus) == 10  
  
 #тест минуса  
 def testminus(self,minus):  
 assert Calculate.calc(\*minus) == 0  
  
 #тест умножения  
 def testumnozhenine(self,umnozh):  
 assert Calculate.calc(\*umnozh) == 20  
  
 #тест деления  
 def testdelenie(self,delen):  
 assert Calculate.calc(\*delen) == 1  
  
 def testerror(self, error):  
 with pytest.raises(TypeError):  
 Calculate.calc(error)

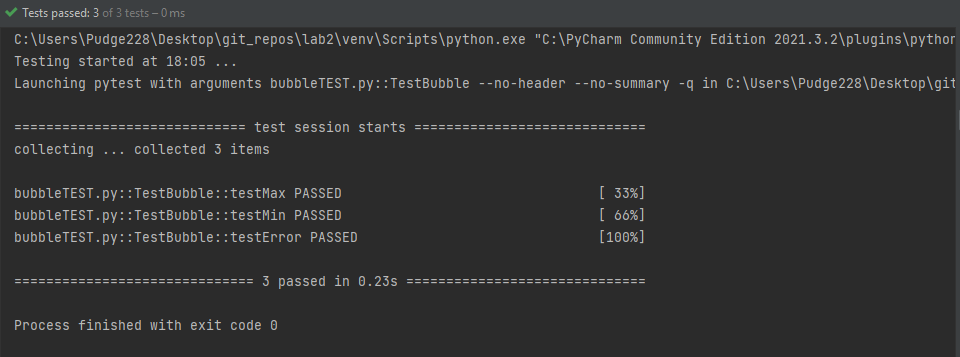
Результаты тестов калькулятора:



Класс тестов сортировки:

# тест сортировки пузырьком  
import pytest  
import Calculate  
  
class TestBubble:  
  
 #сортировка в порядке возрастания  
 @pytest.fixture()  
 def toMax(self):  
 return 1  
  
 #сортировка в порядке убывания  
 @pytest.fixture()  
 def toMin(self):  
 return 2  
  
 # запуск с ошибкой  
 @pytest.fixture()  
 def toError(self):  
 return 3  
  
 #тест сортировки в порядке возрастания  
 def testMax(self, toMax):  
 assert Calculate.bubble(toMax) == [1,2,3,4,5,6,7,8]  
  
 #тест сортировки в порядке убывания  
 def testMin(self, toMin):  
 assert Calculate.bubble(toMin) == [8,7,6,5,4,3,2,1]  
  
 #тест сортировки с ошибкой  
 def testMin(self, toError):  
 assert Calculate.bubble(toError) == -1

Результаты тестов сортировки:



Класс тестов вычисления чисел Фибоначчи:

# Тест чисел Фибоначчи  
import pytest  
import Calculate  
class TestFibanacci:  
 #тест с n = 10  
 def testOne(self):  
 assert Calculate.Fibonacci(10) == [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]  
  
 # тест c n = 0, вызывает исключение IndexError.  
 def testTwo(self):  
 with pytest.raises(IndexError):  
 Calculate.Fibonacci(0)

Результат тестов вычисления чисел Фибоначчи:

