# OOPython

## Задача 4. Функциональные пространства

### Задание

#### Определение классов

**Часть 1**

* создать классы для представления 3-х различных нормированных пространств со следующими правилами вычисления норм:

* +  (пространство )
  +  (пространство )
  +  (пространство )
* создать классы для представления 3-х различных метрических пространств, с правилами вычисления метрик, порождаемых нормами соответствующих нормированных пространств , , 

**Часть 2**

* создать класс для представления 1 предгильбертова пространства  со следующим правилом вычисления скалярного произведения: 
* создать класс для представления 1 нормированного пространства  с нормой, порождаемой скалярным произведением пространства 
* создать класс для представления 1 метрического пространства  с метрикой, порождаемой нормой пространства 

Минимизировать суммарное число строк кода с помощью наследования.

#### Использование классов

Протестировать работу классов на следующих функциях:

* 
* 
* 
* 

1. в нормированных пространствах вычислить нормы всех функций
2. в соответствующих метрических пространствах вычислить попарные расстояния между всеми функциями (на отрезке )
3. в пространстве  вычислить попарные скалярные произведения между всеми функциями
4. в пространстве  попарно вычислить углы между всеми функциями

### Реализация

#### Поиск максимума функции

Производить на множестве значений функции, заданных в узлах равномерной сетки на отрезке. Шаг сетки .

#### Вычисление значений производных

Для 1-й производной использовать формулу центральной разности (2-й порядок точности). Для вычислений использовать соответствующий класс, реализованный Вами в **Задаче 2.** Вычисление значений 2-й производной производить с помощью этого же класса. Шаг численного дифференцирования .

#### Вычисление интеграла

Производить по формуле трапеций (2-й порядок точности). Для вычислений использовать соответствующий класс, реализованный Вами в **Задаче 3.** Шаг численного интегрирования .

*Примечание 1:* для поиска максимума функции на отрезке разрешено использовать функцию **numpy.amax(array).**

*Примечание 2:*«закрыть глаза»на использование значений функций вне отрезка **** в процессе вычисления производных.