# OOPython

## Задача 5. Численное решение ОДУ

### Введение

Будем рассматривать следующие методы численного решения ОДУ:

1. явный метод Эйлера (явный, 1-й порядок аппроксимации; реализован в **lecture\_8\_scalar\_ode.ipynb**)
2. метод Хойна (явный, 2-й порядок аппроксимации; реализован в **lecture\_8\_scalar\_ode.ipynb**)
3. метод Рунге-Кутты (явный, 4-й порядок аппроксимации; реализован в **lecture\_8\_scalar\_ode.ipynb**)
4. метод трапеций (неявный, 2-й порядок аппроксимации).

### Задание

#### Определение классов

Для каждого из методов (1)-(4) реализовать соответствующий класс *MethodName*, минимизировав суммарное число строк кода с помощью наследования. Каждый класс должен включать в себя как минимум следующее:

**Поля**:

* функция правой части ОДУ
* начальное условие
* множество точек сетки и ее параметры: кол-во точек, отрезков, шаг сетки
* массив для хранения значений численного решения
* начальный и конечный моменты времени.

**Методы:**

* конструктор: задать правую часть, начальное условие, параметры сетки
* решить ОДУ («timestepping» - цикл по точкам сетки)
* построить график численного решения.

#### Использование классов

Каждым из методов произвести численное решение логистического уравнения и построить графики полученных решений (в одном графическом окне).

Параметры функции правой части уравнения:

Параметры расчетной сетки:

* число отрезков разбиения

Начальное условие:

### Примечания по реализации

#### Метод трапеций (неявный, 2-й порядок аппроксимации)

Формула для проведения вычислений:

Для решения получившегося нелинейного уравнения относительно будем использовать итерационный метод Ньютона:

Для ускорения сходимости начальное приближение можно рассчитывать по следующим формулам:

Для приближенного вычисления производной использовать формулу центральной разности 2-го порядка точности:

Решение нелинейного уравнения с помощью метода Ньютона можно оформить как виде метода класса, так и в виде отдельной функции. При вычислении производной задействовать функторы, реализованные в **Задаче 2**. Критерий остановки итераций: