

ЗАДАНИЕ 2

Напишите библиотеку, с помощью которой можно будет вычислять значения и производную функции одной переменной. *Основные функции:*

- тождественная функция,
- вещественные константы,
- степенная функция,
- экспонента,
- полином.

Библиотека должна поддерживать выражения, которые собираются из таких объектов и арифметических операций над ними, поддерживаемые путем переопределения операторов. При этом необходимо генерировать исключения `std::logic_error` в случае арифметических операций функций с другими типами. Также у *основных* функций должны быть поддержаны методы `ToString`, возвращающие строковое представление функции.

1. Реализуйте иерархию классов *основных* функций, базовый класс `TFunction` в которой хранит функторы для вычисления функции и ее производной, а также фабрику по созданию таких *основных* функциональных объектов. Создание объекта должно задаваться параметром `type`, который принимает значения из множества `ident`, `const`, `power`, `exp`, `polynomial`, ..., а также необходимым параметром для определения функциональности, например, для констант или функции `power`.

```
// один из вариантов, схематично:
std::vector<TFunctionPtr> cont;
auto f = funcFactory.Create("power", 2); // TPowerFunc x^2
cont.push_back(f);
auto g = funcFactory.Create("polynomial", {1, 2, 3}); // TPolinomial 1 + 2*x^2 + 3*x^3
cont.push_back(g);
for (const auto ptr : cont) {
    std::cout << ptr->ToString() << " for x = 10 is " << (*ptr)(10) << std::endl;
}
auto p = *f + *g;
p.GetDeriv(1); // 15
auto h = *f + "abc"; // std::logic_error
f->GetDeriv(3); // 6
```

2. В данной библиотеке реализуйте функцию, которая принимает на вход арифметическое выражение $f(x)$ с одной переменной, и находит корень уравнения $f(x) = 0$ методом градиентного спуска. Проверять сходимость не надо. Число итераций может быть параметром функции.

3. С использованием `gmock/gtest` напишите тесты на функции библиотеки и основные классы `TFunction`. Протестируйте создание объектов, генерацию ошибок, вычисления функций и производных. Напишите несколько тестов на градиентный спуск (например, решение квадратного уравнения).

Замечания.

- (1) Метод `ToString` можно поддержать только у *основных* функций.
- (2) Сложение двух полиномов не требует специальных представлений со сложением соответствующих коэффициентов.
- (3) Производную функции нужно уметь вычислять в точке, необходимости делать ее как `TFunction` нет.
- (4) Необходимо продумать иерархию классов *основных* функций, чтобы правильно их приводить к строкам и создавать.

Решение задачи нужно прислать (предпочтительнее ссылка на github, и т.п.) на `vkopovodov@gmail.com`. Решение должно быть оформлено в виде компилирующегося кода.

Срок сдачи задания — 7 декабря 2019 г. 23:59.