

Задание №3 практикума по C++

(в поддержку курса «Проектирование и программирование на C++»)

Напишите библиотеку, с помощью которой можно будет вычислять значения и производную функции одной переменной. *Основные* функции, которые должны представляться в виде объектов:

- тождественная функция,
- вещественная константа,
- степенная функция,
- экспонента,
- полином.

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

1. Реализуйте иерархию классов *основных* функций, базовый класс `TFunction` из которой хранит функторы для вычисления функции и ее производной, а также фабрику по созданию таких *основных* функциональных объектов. Создание объекта должно задаваться строковым параметром `type`, который принимает значения из множества `{"ident", "const", "power", "exp", "polynomial"}`, а также параметром, необходимым для определения функции выбранного типа, например: значение вещественной константы, показатель степенной функции.

```
// один из вариантов, схематично:
std::vector<TFunctionPtr> cont;
auto f = funcFactory.Create("power", 2); // PowerFunc x^2
cont.push_back(f);
auto g =
    funcFactory.Create("polynomial", {7, 3, 15}); // TPolynomial 7 + 3*x^2 + 15*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 градиентного спуска должно быть параметром функции.

3. С использованием `gmock/gtest` напишите тесты для классов библиотеки, поддерживающих *основные* функции, и для операций составления арифметических выражений. Протестируйте создание объектов, генерацию ошибок, вычисление функций и производных, формирование строкового представления. Напишите несколько тестов на градиентный спуск (например, решение квадратного уравнения).

Замечания.

- (1) Метод `ToString` достаточно 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 реализовывать производную как `TFunction` нет.
- (4) Необходимо продумать иерархию классов *основных* функций, чтобы правильно их создавать и приводить к строковому виду.

- (5) О паттерне «фабрика» можно почитать здесь:
https://mk.cs.msu.ru/images/5/57/Lecture_pbscpp_8.pdf
- (6) Вопросы по заданию следует направлять Владимиру Коноводову (vkonovodov@gmail.com), копия – Василию Балашову (hbd@cs.msu.su).