

Первое задание по практикуму для самостоятельного программирования дома

Напишите программу, в которой определяются классы «переменная», «функция» и другие вспомогательные классы. Предполагается, что функции могут зависеть не более чем от одного аргумента и представляют собой суперпозиции операций сложения, вычитания, умножения, а также элементарных функций синуса, косинуса, логарифма, экспоненты, арксинуса. Для классов, участвующих в программе, должны быть перегружены соответствующие функции и операции, так чтобы использующий данные классы мог легко строить суперпозиции функций и вычислять их значения следующим образом:

```
// Define variables x and y that are correspondingly named:
Variable x("x"), y("y");
Function f = 3 * x * x - 2 * x;
std::cout << f; // Print some text representation of f
std::cout << f(x); // Print the same representation of f
std::cout << f(2 + x * x); // Print text representation of f(2 + x * x)
std::cout << f(y); // Print representation of f in which "x" is replaced by "y"
Function g = sin(x) * sin(x) + cos(x) * cos(x);
Function h = exp(y) + log(1 + y * y);
h = h + 2.5 * exp(h(y * y)) * cos(sin(h)) * y;
h = h + g; // Error: rhs depends on two arguments (both x and y)
h = h(x) + g(x); // OK, now h by default depends on x
std::cout << f(1) << ' ' << 2 * g(0) << ' ' << h(0) + 1; // Print 1 2 3
f = asin(x);
g = exp(x);
std::cout << 6 * f(0.5) << ' ' << g(1); // Print constants  $\pi$  and  $e$ 
h = 6 * f(0.5 * g(x) + x * x * g(f(x)));
// Print constant  $\pi$  with 100 significant digits:
std::cout << std::setprecision(100) << h(0);
double x = h(0); // h(0) is calculated using std::asin and std::exp
std::cout << x; // Standard print of the variable x
Function huge = 1;
for (int i = 0; i < 1000; ++i) {
    huge = huge * 10;
}
Function sum = huge + 1;
std::cout << std::setprecision(1002) << sum(0); // Print  $10^{1000} + 1$  as a number
std::cout << std::setprecision(2) << (sum - huge)(0); // Print 1
```

Можно считать, что для любого ненулевого вещественного числа $x = M \cdot 10^p$, $M \in [1, 10)$, значение его порядка p может быть представлено типом `int`. Число значащих цифр мантиссы M должно быть вычислено исходя из точности, с которой нужно вычислить итоговое значение. При выводе символьные представления функций можно не упрощать. Можно считать, что аргументы логарифма лежат в интервале $(0, 2)$, аргументы арксинуса лежат в интервале $(-1, 1)$. В функцию `main` поместите свой код, показывающий работоспособность программы.

Требования:

1. Все манипуляторы `std::ostream`, кроме `std::setprecision`, можно игнорировать. При их игнорировании вещественные числа нужно выводить в формате с фиксированной точкой. Незначащие нули в конце дробной части числа могут быть оставлены или опущены по желанию.
2. При выводе значения функции с заданной точностью допускается ошибка не более чем на 1 в последнем разряде. Например, при выводе числа π с 5 значащими цифрами могут быть использованы *только* следующие комбинации значащих цифр: 3.1414, 3.1415, 3.1416.
3. Если для некоторого класса в программе автоматически генерируемые конструктор копирования, операция присваивания или деструктор работают некорректно, они должны быть заменены на пользовательские.
4. Запрещается пользоваться STL.
5. При возникновении ошибки нужно сообщить об этом пользователю. Форма сообщения может быть произвольной, например, через компилятор или с помощью вывода на экран.
6. Программа должна быть разбита на несколько модулей. Один из них должен представлять .cpp-файл, содержащий функцию `main`, другие модули должны быть представлены в виде пар .hpp- и .cpp-файлов (части имён до «.» совпадают), первый из которых содержит объявления классов и функций, второй — их описания. Все описываемые классы и функции должны находиться в некотором (не внешнем) пространстве имён, .hpp-файлы должны быть защищены от повторного включения (`#ifndef - #define; #pragma`).

Срок сдачи: 23:59, 18.03.2022.