# ФИТ НГУ, курс ООП, осенний семестр

# **Задание №2. (25 баллов) Шаблон проектирования «фабричный метод», модульное тестирование (unit testing).**

## ***Часть 1. (15 баллов).***

Написать стековый калькулятор, который принимает в качестве аргумента командной строки имя файла, содержащего команды. Если аргумента нет, то использовать стандартный поток ввода для чтения команд. Использовать вещественные числа.

Реализовать следующий набор команд:

* **#** - строка с комментарием.
* **POP**, **PUSH** — снять/положить число со/на стек(а).
* **+ , - , \* , /, SQRT –** арифметические операции. Используют один или два верхних элемента стека, изымают их из стека, помещая результат назад
* **PRINT** — печать верхнего элемента стека (без удаления).
* **DEFINE** — задать значение параметра. В дальнейшем везде использовать вместо параметра это значение.

Пример (должно вывести 2):   
*DEFINE a 4  
PUSH a  
SQRT   
PRINT*

## ***Часть 2. (10 баллов).***

1. Реализовать набор модульных тестов, покрывающих функционал калькулятора.

### **Методические указания:**

* Создание команд рекомендуется реализовать посредством шаблона проектирования «фабричный метод» ([http://ru.wikipedia.org/wiki/Фабричный\_метод\_(шаблон\_проектирования](http://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B1%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)[)](http://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B1%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F))).
* Аргументы команде (тем у которых есть аргументы) на исполнение можно передавать в виде списка объектов, команда сама должна уметь интерпретировать свои аргументы
* Содержимое стека и список (лучше ассоциативный контейнер std::map<std::string, double>) определенных именованных параметров передавать команде в виде специального объекта — контекста исполнения
* Разработать иерархию исключений, которые будут выбрасывать команды при исполнении. В случае возникновения исключения — выводить информацию об ошибке и продолжать исполнение программы (из файла или команд вводимых с консоли)
* Для реализации модульных тестов использовать Google Test Framework (<https://ru.wikipedia.org/wiki/Google_C%2B%2B_Testing_Framework>, <https://www.ibm.com/developerworks/aix/library/au-googletestingframework.html>)