# 2022

# Клещенко В.Д. z33431

# ЛР #2: [C++ & UNIX]: C++ BUILD / IF / LOOP, PYTHON

## Цель

Познакомить студента с принципами компиляции исходного кода. Составить программу с использованием циклов, условий и функций. Сравнить быстродействие между C++ и Python. Ознакомление с типами данных.

## Задача

1. **[С++ EXPRESSION] Создать и скомпилировать программу на C++**

Результат сборки (компиляции) сохранять в папку build. Папку build сделать игнорируемой для GIT. Программа должна получать на вход число – это количество итераций для выполнения расчета. В рамках итерации выполнять следующее вычисление: x^2-x^2+x\*4-x\*5+x+x. Вычисление выполнять в виде отдельной от main функции, которая будет вызвана циклически из main. Фиксировать время выполнения программы, затрачиваемое на расчет выражения n раз (n задается в консоли перед вычислением). Предусмотреть дополнительный цикл на повторную итерацию запуска программы вычислений. Если было введено не число, то завершить выполнение программы.

1. **[PYTHON EXPRESSION] Создать и скомпилировать программу на Python 3**

Результат сборки (компиляции) сохранять в папку build. Папку build сделать игнорируемой для GIT. Программа должна получать на вход число – это количество итераций для выполнения расчета. В рамках итерации выполнять следующее вычисление: x^2-x^2+x\*4-x\*5+x+x. Вычисление выполнять в виде отдельной от main функции, которая будет вызвана циклически из main. Фиксировать время выполнения программы, затрачиваемое на расчет выражения n раз (n задается в консоли перед вычислением). Предусмотреть дополнительный цикл на повторную итерацию запуска программы вычислений. Если было введено не число, то завершить выполнение программы.

1. **[SAVE] Результат всех вышеперечисленных шагов сохранить в репозиторий (+ отчет по данной ЛР в папку doc)**

Фиксацию ревизий производить строго через ветку dev. С помощью скриптов накатить ревизии на stg и на prd. Скрипты разместить в корне репозитория. Также создать скрипты по возврату к виду текущей ревизии (даже если в папке имеются несохраненные изменения + новые файлы).

# Решение

1. Ниже представлена программа на с++ которая вычисляет некоторое выражение определенное количество раз, а также фиксирует время выполнения. Также есть функция возведения в степень, так как нельзя использовать встроенную. Проверка на число происходит в условии цикла при присвоении значения переменной.

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст, внутренний, монитор, компьютер

Автоматически созданное описание Изображение выглядит как текст, монитор, внутренний, электроника

Автоматически созданное описание

2.

Ниже представлена программа на python которая производит те же действия. Проверка на число происходит в условии цикла но не при присвоении значения переменной, а с помощью встроенной функции проверки на отсутствие «букв».

Изображение выглядит как текст, снимок экрана, монитор, электроника

Автоматически созданное описание Изображение выглядит как текст, монитор, снимок экрана, внутренний

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

3.

Отменяет все индексированные и неиндексированные изменения в рабочей области, сохраняя их в карман (stash):

git stash save

--keep index позволяет не только

Изображение выглядит как текст, компьютер, ноутбук, внутренний

Автоматически созданное описаниеИзображение выглядит как текст, компьютер, ноутбук, снимок экрана

Автоматически созданное описание

# Заключение

В результате лабораторной работы было произведено знакомство с принципами компиляции исходного кода, ознакомление с типами данных, cоставлена программа с использованием циклов, условий и функций, сравнено быстродействие между C++ и Python: C++ ожидаемо работает быстрее Python это становится заметно на 100 000 итерациях: 0.003 c против 153 с – разница в 50 раз. На 1 000 000 итераций – разница на порядок.