

# Семинар: Основы CMake. Домашнее задание.

## Задача 1. Простой проект

В папке `0simple_project` содержится исходный код простой программы, которая создаёт изображение. Создайте проект Cmake, который бы собирал этот проект.

## Задача 2. Библиотека

В папке `1creating_library` содержится код маленькой библиотеки для работы с изображениями. Создайте проект Cmake, который бы создавал статическую библиотеку.

## Задача 3. Использование библиотеки

Перенесите файл библиотеки из прошлой задачи в папку `2linking_library`. Создайте проект Cmake, бы подключал статическую библиотеку `image` к файлу `main.cpp` и собирал исполняемый файл.

## Задача 4. Работа со списками

В файле `3list/animals.cmake` содержится переменная `animals`, содержащая список животных. Напишите скрипты Cmake, которые будут делать следующее:

- Печатать элементы списка на экран с помощью цикла. Каждое название животного в новой строке.
- Печатать количество элементов списка
- Печатать только те элементы списка, в который больше 4-х символов
- Печатать только те элементы списка, которые начинаются на *S*
- Сортировать список и печатать его
- Печатать элементы списка на экран в одну строку через пробел.
- Создавать файл `data.txt` и сохранять в него список.
- Создавать папку `animal_files` и создавать в неё файлы с названиями элементов списка с расширением `.txt` (то есть `cat.txt`, `mouse.txt` и т. д.).

## Задача 5. Выбор файла исходного кода

В папке `4choosing_source/src` лежат три файла исходного кода. Каждый из файлов может компилироваться в исполняемый файл. Вам нужно дописать файл `4choosing_source/CMakeLists.txt` так, чтобы можно было выбирать какой файл будет компилироваться во время конфигурации проекта в зависимости от значения переменной `ANIMAL_TYPE`. То есть если мы перейдём в папку `build` и запустим:

```
cmake -DANIMAL_TYPE=Cat ..
cmake --build .
```

то при сборке должен компилироваться файла `src/cat.cpp`, а получившийся исполняемый файл должен печатать на экран слово `Cat`. Если же мы запустим:

```
cmake -DANIMAL_TYPE=Mouse ..
cmake --build .
```

то при сборке должен компилироваться файла `src/mouse.cpp`, а получившийся исполняемый файл должен печатать на экран слово `Mouse`. Если же мы не установим значение переменной `ANIMAL_TYPE`, либо установим её значение не равное одному из значений: `Cat`, `Dog`, `Mouse`, то попытка конфигурации проекта должна приводить к фатальной ошибке.

## Задача 6. Функция печати переменной

На языке Cmake напишите функцию `print`, которая должна будет принимать на вход название переменной и печатать название переменной и её значение. Например, следующий код:

```
set(VAR "Cats and Dogs")
print(VAR)
```

должен напечатать на экран

```
VAR = "Cats and Dogs"
```

## Задача 7. Передача в функцию

На языке CMake написана функция `func`, которая печатает количество переданных ей аргументов и значения этих аргументов:

```
function(func)
    message("Number of arguments = " $ARGC)
    message("Arguments:")
    foreach(arg IN LISTS ARGV)
        message($arg)
    endforeach()
    message("-----")
endfunction()
```

Предположим, что есть переменная `X` определённая так:

```
set(X "Tiger;Lion;Elephant")
```

Что напечатает данная функция, если вызывать её следующим образом:

1. `func()`
2. `func(Cat)`
3. `func(Cat Mouse Dog)`
4. `func("Cat Mouse Dog")`
5. `func(Cat;Mouse;Dog)`
6. `func("Cat;Mouse;Dog")`
7. `func(X)`
8. `func($X)`
9. `func("$X")`
10. `func("Cat;Mouse;Dog" "$X")`

Для ответа на этот вопрос запишите результаты по каждому пункт в один текстовый файл.

## Задача 8. Сортировка пузырьком

На языке CMake напишите функцию которая будет принимать список и возвращать отсортированную версию этого списка.