

Федеральное государственное автономное
образовательное учреждение высшего
образования
«Национальный исследовательский университет
ИТМО»

Факультет Информационных технологий и программирования

Работа: Лабораторная работа № 5
IDE

Выполнила студентка группы №М3103
Кравченкова Елизавета Александровна

Проверил:
Повышев Владислав Вячеславович

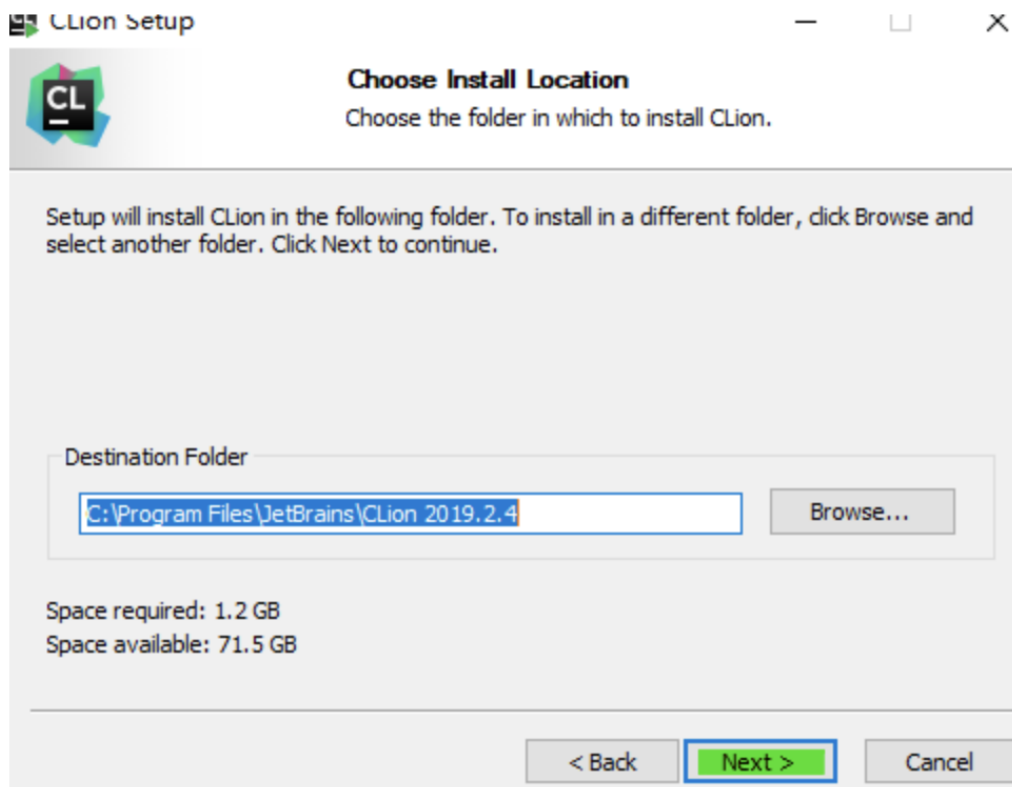
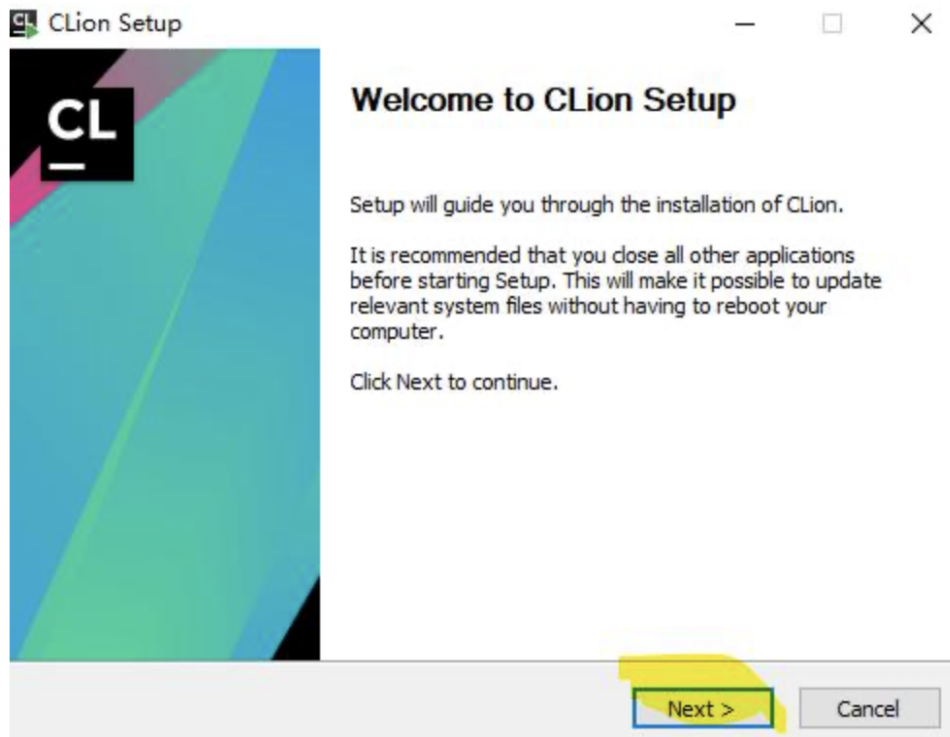
Санкт-Петербург
2022 г.

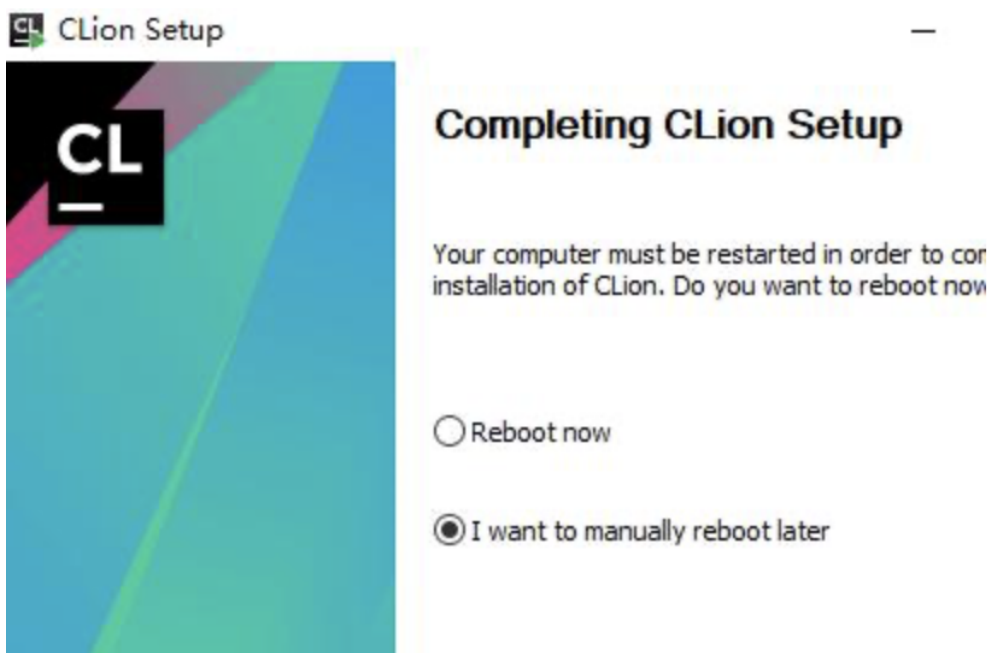
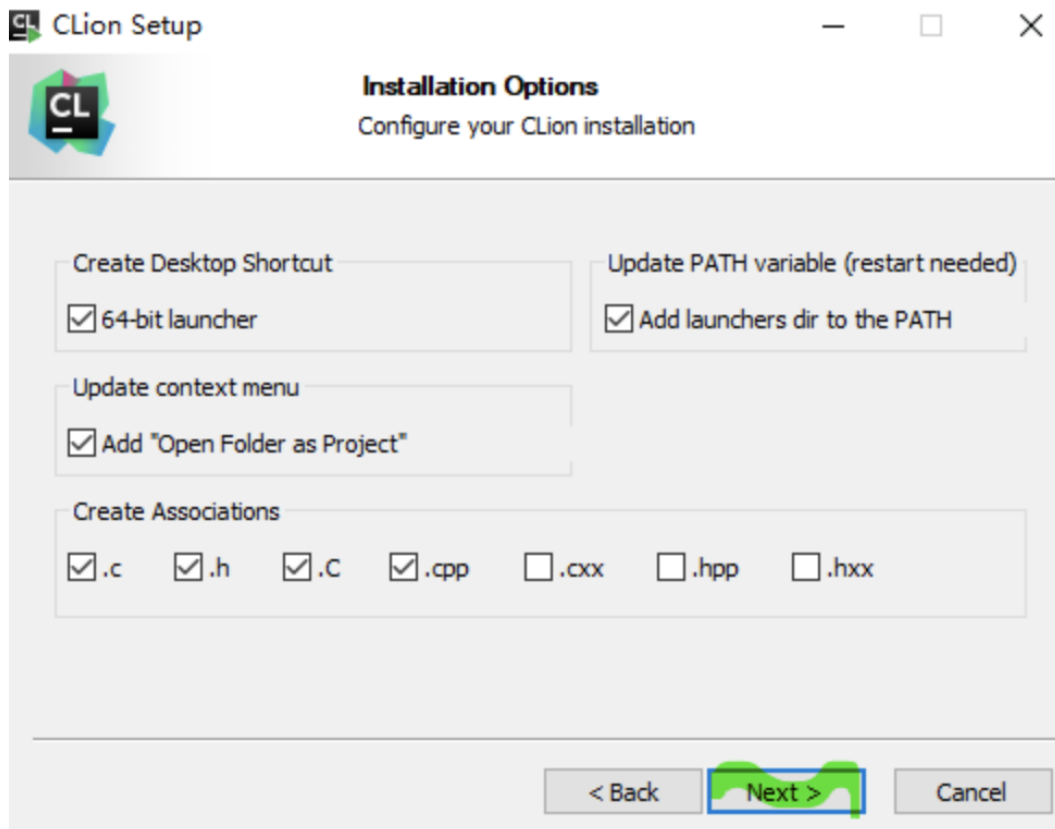
1. Установка CLion

Переходим по данной ссылке и скачиваем

<https://www.jetbrains.com/clion/download/#section=mac>

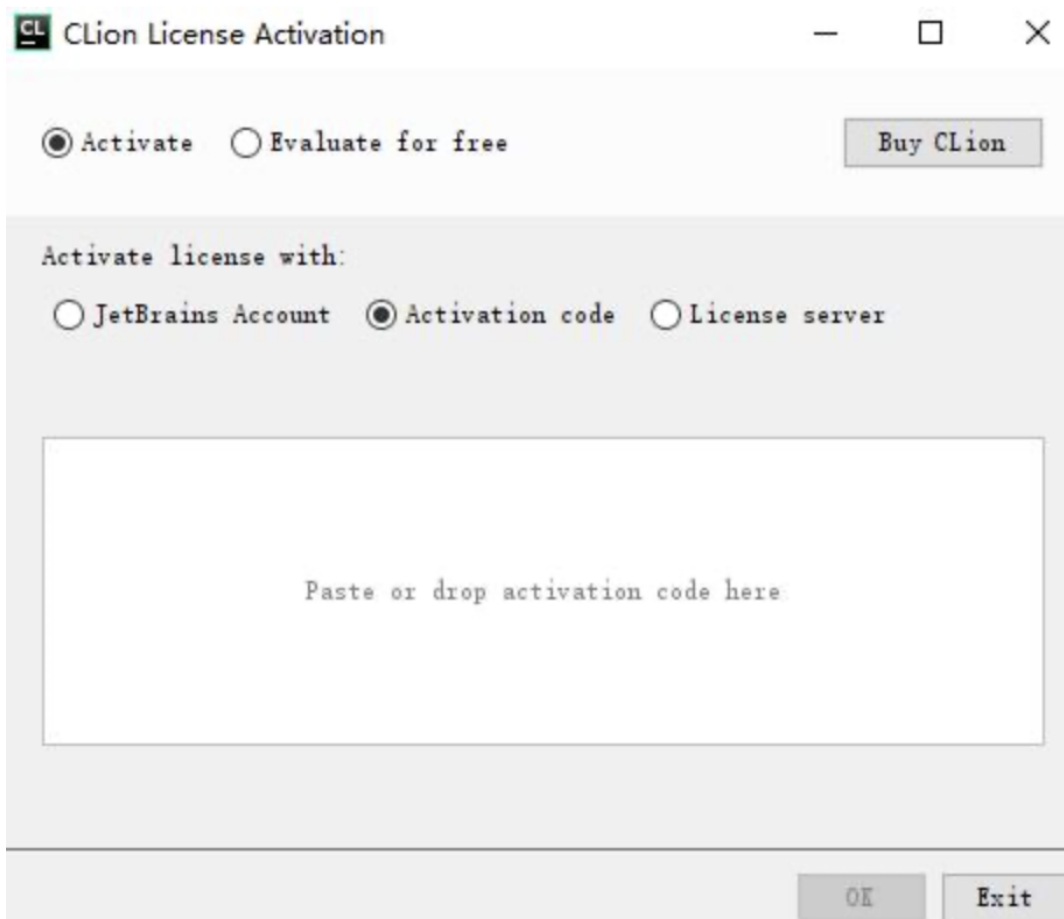
После загрузки соглашаемся со стандартными настройками, нажимаем везде далее





2. Активация ключа

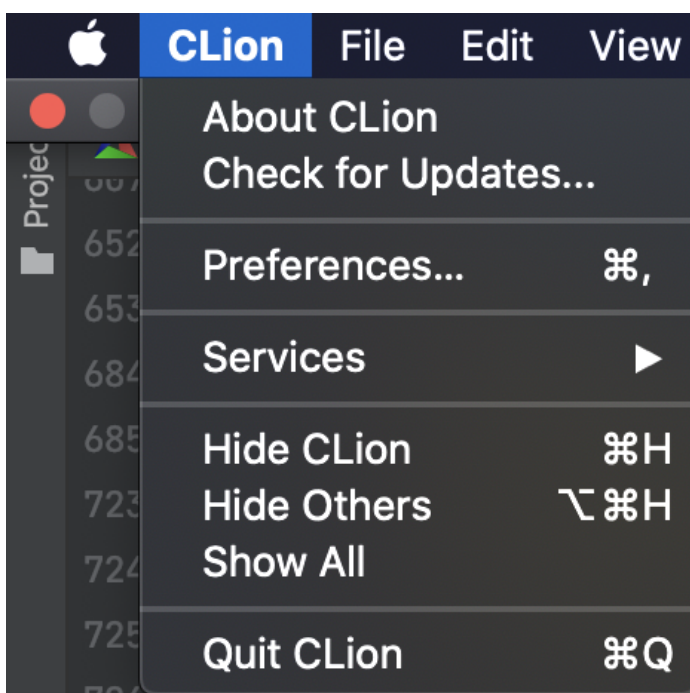
Пишем ключ активации, который можно получить на сайте <https://www.jetbrains.com>



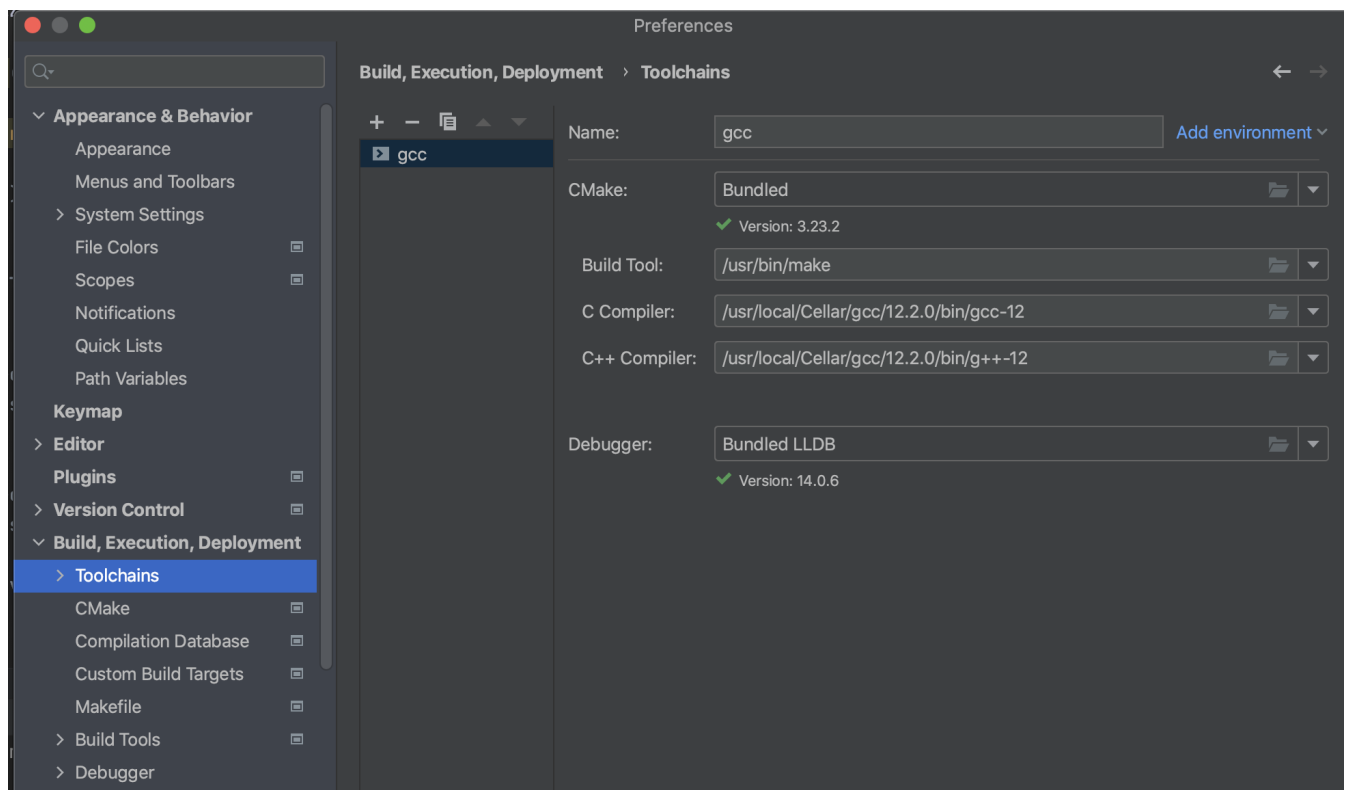
3. Установка gcc

Вводим в консоль `brew install gcc`

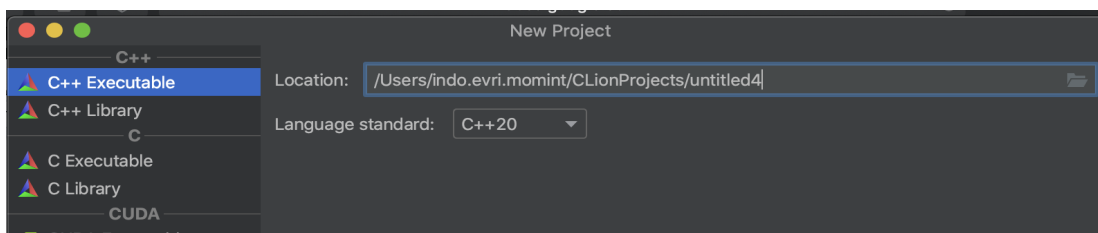
Заходим в CLion в preferences



В разделе Toolchains в поле C Compiler и C++ Compiler записываем путь к компилятору



4. Настройка IDE завершена. Создаем проект.



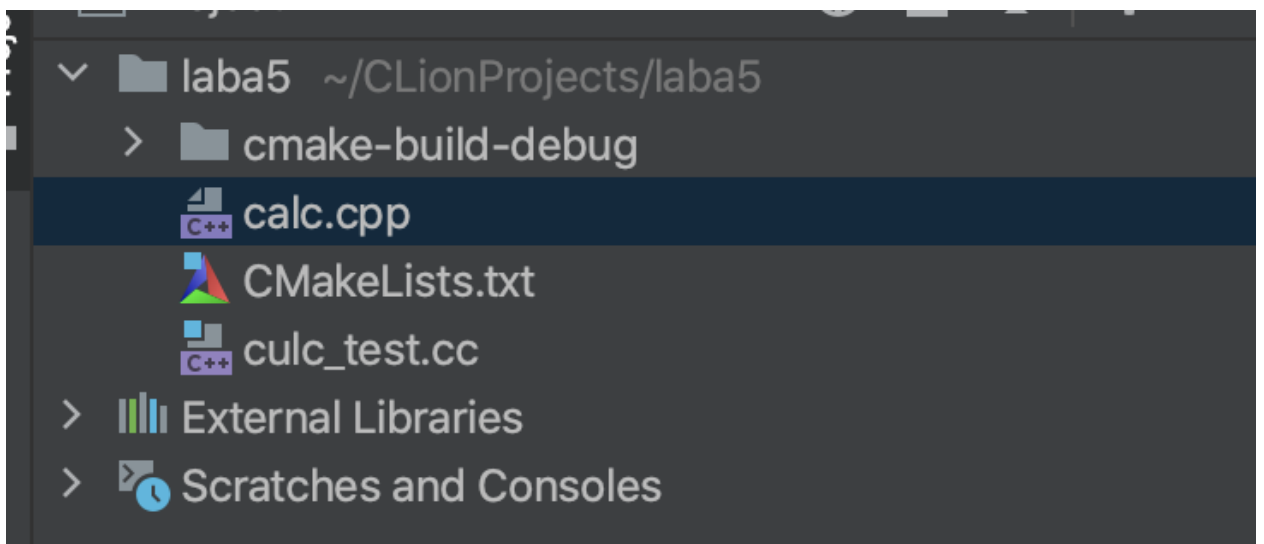
5. Пишем калькулятор

```
int64_t Culk(int64_t x, int64_t y, char ch){  
    if (ch == '+'){...}  
    else if (ch == '-'){  
        return x - y;  
    }  
    else if (ch == '*'){  
        return x * y;  
    }  
    else if (ch == '/'){  
        return x / y;  
    }  
    else{  
        std::cerr << "Введена неверная опция" << std::endl;  
        return 0;  
    }  
}
```

```
void FriendInterface() {  
    int64_t x;  
    int64_t y;  
    char ch;  
    std::cout << "Привет, это милый калькулятор" << std::endl;  
    std::cout << "Введи @ , чтобы прекратить его работу" << std::endl;  
  
    while (true) {  
        std::cout << "Введите первое число" << std::endl;  
        std::cin >> x;  
        std::cout << "Введите второе число" << std::endl;  
        std::cin >> y;  
  
        std::cout << "Введите операцию(+*/) " << std::endl;  
        std::cin >> ch;  
        if (ch == '@') {  
            break;  
        }  
        int64_t ans = Culk(x, y, ch);  
        std::cout << ans;  
    }  
}
```

6. Подключим Google tests


Для этого при таком дереве файлов, запишем в CMakeLists следующее













```
CMakeLists.txt x culc_test.cc x
1 cmake_minimum_required(VERSION 3.14)
2 project(my_project)
3
4 # GoogleTest requires at least C++14
5 set(CMAKE_CXX_STANDARD 20)
6
7 include(FetchContent)
8 FetchContent_Declare(
9     googletest
10     URL https://github.com/google/googletest/archive/03597a01ee50ed33e9dfd640b249b4be3799d395.zip
11 )
12
13 # For Windows: Prevent overriding the parent project's compiler/linker settings
14 set(gtest_force_shared_crt ON CACHE BOOL "" FORCE)
15 FetchContent_MakeAvailable(googletest)
16
17 enable_testing()
18
19 add_executable(
20     culc_test
21     culc_test.cc
22 )
23 target_link_libraries(
24     culc_test
25     GTest::gtest_main
26 )
27
28 include(GoogleTest)
29 gtest_discover_tests(culc_test)
```



7. Написание тестов

```
1  #include <gtest/gtest.h>
2  #include "calc.cpp"
3
4  TEST(SumTest, Basic) {
5      EXPECT_EQ(CuLk( x: 6, y: 7, ch: '+'), 13);
6      EXPECT_EQ(CuLk( x: 5, y: 5, ch: '+'), 10);
7  }
8
9  TEST(SumTest, Zero) {
10     EXPECT_EQ(CuLk( x: 0, y: 7, ch: '+'), 7);
11     EXPECT_EQ(CuLk( x: 2, y: -2, ch: '+'), 0);
12 }
13
14 TEST(SumTest, Negative) {
15     EXPECT_EQ(CuLk( x: -6, y: 7, ch: '+'), 1);
16     EXPECT_EQ(CuLk( x: -6, y: -10, ch: '+'), -16);
17 }
18
19 TEST(SumTest, Big) {
20     EXPECT_EQ(CuLk( x: 6000, y: 20, ch: '+'), 6020);
21     EXPECT_EQ(CuLk( x: 444444, y: 55555, ch: '+'), 499999);
22 }
23
24 TEST(DiffTest, Basic) {
25     EXPECT_EQ(CuLk( x: 6, y: 7, ch: '-'), -1);
26     EXPECT_EQ(CuLk( x: 2, y: -2, ch: '-'), 4);
27 }
28
29 TEST(DiffTest, Zero) {
30     EXPECT_EQ(CuLk( x: 0, y: 7, ch: '-'), -7);
31     EXPECT_EQ(CuLk( x: 2, y: 2, ch: '-'), 0);
32 }
33
34 TEST(DiffTest, Negative) {
35     EXPECT_EQ(CuLk( x: -6, y: 7, ch: '-'), -13);
36     EXPECT_EQ(CuLk( x: -6, y: -10, ch: '-'), 4);
37 }
```

Run:  culc_test x

         >>  Tests passed: 16 of 16 tests – 0 ms

Test Results 0 ms /Users/indo.evri.momint/CLionProjects/lab5/cmake-b
Testing started at 02:09 ...
Running main() from /Users/indo.evri.momint/CLionPr
Process finished with exit code 0

 culc_test | Debug 

8. Изучим сочетания клавиш в CLion для Mac OS

Выполнение и отладка

-Сборка $\text{⌘}F9$

-Отладка $\text{⌘}D$

-Run $\text{⌘}R$

Комментирование

Для выделенных фрагментов блочное ($\text{⌘}/*$) и однострочное комментирование ($\text{⌘}/$)

Комментарий к одной строке $\text{⌘}/$.

Навигация

Посмотреть реализацию или объявление функции $\text{⌘}Space$ (в окне)

Перейти к объявлению функции ($\text{⌘}B$) или его определению ($\text{⌘}B$)

Поиск нужного класса ($\text{⌘}O$), файла ($\text{⌘}O$) или символа ($\text{⌘}O$) по всему проекту.

Рефакторинг

Отформатировать выделенный блок кода $\text{⌘}L$

Все рефакторинги, доступные в текущем месте $\text{⌘}T$

Rename ($\text{⌘}F6$)

Change Signature ($\text{⌘}F6$) (поможет добавить/удалить параметры функции и поменять их порядок, а также изменить тип результата и имя функции, при этом все использования будут обновлены автоматически)