

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

Факультет Информационных технологий и программирования

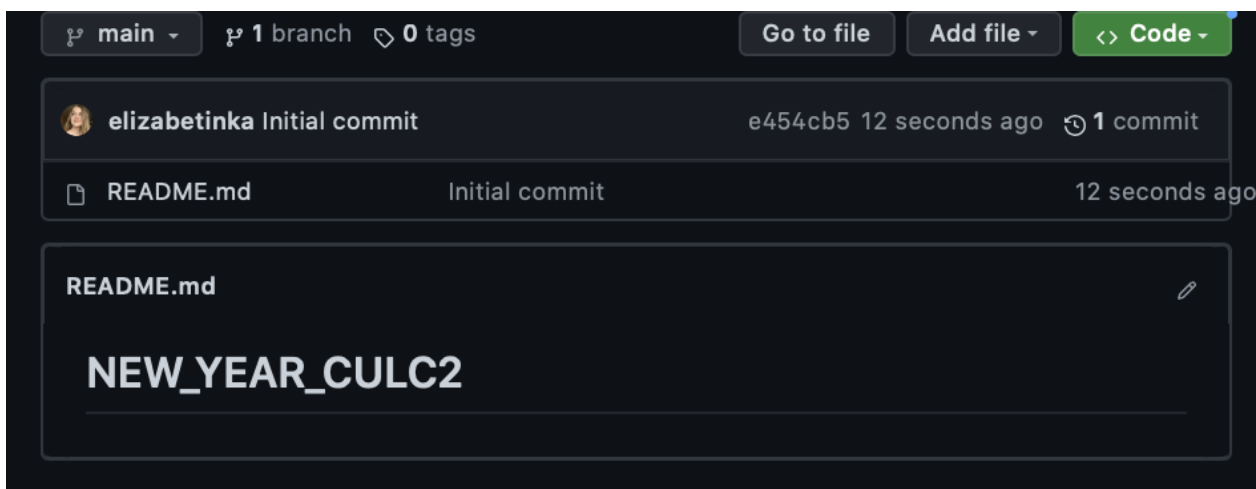
Работа: Лабораторная работа № 6  
Настройка CI CD

Выполнила студентка группы №М3103  
*Кравченкова Елизавета Александровна*

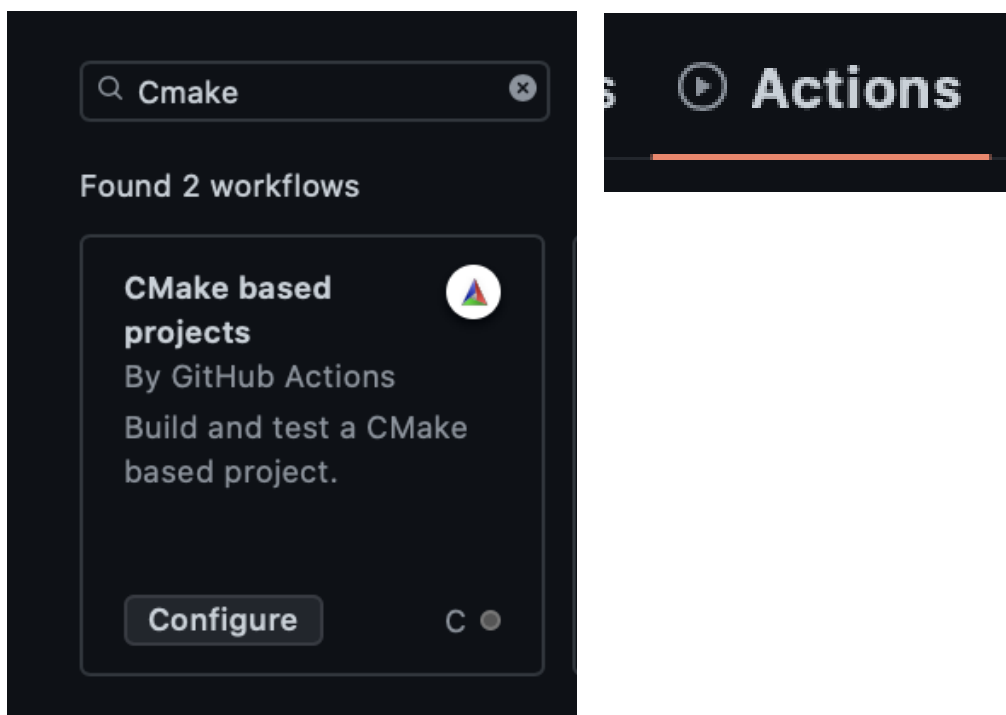
Проверил:  
*Повышев Владислав Вячеславович*

Санкт-Петербург  
2022 г.

1. Создадим репозиторий. Склонируем его и скопируем в него проект из прошлой лабораторной работы. (Калькулятор с Google Tests).



2. Перейдем во вкладку Github Actions. Так как я работаю с CMake выберем



3. Github автоматически создаст Yaml файл, в котором настроены базовые настройки, триггеры, сборка и прогон тестов. Сделаем КОММИТ.

```
NEW_YEAR_CULC2 / .github / workflows / cmake.yml in main
<> Edit file Preview changes Spaces 2 No wrap


1 name: CMake
2
3 on:
4   push:
5     branches: [ "main" ]
6   pull_request:
7     branches: [ "main" ]
8
9 env:
10  # Customize the CMake build type here (Release, Debug, RelWithDebInfo, etc.)
11  BUILD_TYPE: Release
12
13 jobs:
14   build:
15     # The CMake configure and build commands are platform agnostic and should work equally well on Windows or Mac.
16     # You can convert this to a matrix build if you need cross-platform coverage.
17     # See: https://docs.github.com/en/free-pro-team@latest/actions/learn-github-actions/managing-complex-workflows#using-a-build-matrix
18     runs-on: ubuntu-latest
19
20     steps:
21     - uses: actions/checkout@v3
22
23     - name: Configure CMake
24       # Configure CMake in a 'build' subdirectory. `CMAKE_BUILD_TYPE` is only required if you are using a single-configuration generator such as make.
25       # See https://cmake.org/cmake/help/latest/variable/CMAKE_BUILD_TYPE.html?highlight=cmake_build_type
26       run: cmake -B ${github.workspace}/build -DCMAKE_BUILD_TYPE=${env.BUILD_TYPE}
27
28     - name: Build
29       # Build your program with the given configuration
30       run: cmake --build ${github.workspace}/build --config ${env.BUILD_TYPE}
31
32     - name: Test
33       working-directory: ${github.workspace}/build
34       # Execute tests defined by the CMake configuration.
35       # See https://cmake.org/cmake/help/latest/manual/ctest.1.html for more detail
36       run: ctest -C ${env.BUILD_TYPE}
```


Cancel changes Start commit -


Marketplace Documentation


Search Marketplace for Actions


Featured Actions





**Cache**  
By actions   
Cache artifacts like dependencies and build outputs to improve workflow execution time  
☆ 3.3k




**Upload a Build Artifact**  
By actions   
Upload a build artifact that can be used by subsequent workflow steps  
☆ 2k



**Download a Build Artifact**  
By actions   
Download a build artifact that was previously uploaded in the workflow by the upload-artifact action  
☆ 800



**Setup .NET Core SDK**  
By actions   
Used to build and publish .NET source. Set up a specific version of the .NET and  
☆ 640

4. После коммита зайдём в раздел Actions и там увидим, что началась автоматическая сборка проекта. Заметим, что сборка прошла успешно, все тесты пройдены

#### build

succeeded 1 minute ago in 27s

- > Set up job
- > Run actions/checkout@v3
- > Configure CMake
- > Build

#### Test

```
1 ▶ Run ctest -C Release
6 Test project /home/runner/work/NEW_YEAR_CULC2/NEW_YEAR_CULC2/build
7   Start 1: SumTest.Basic
8   1/16 Test #1: SumTest.Basic ..... Passed    0.00 sec
9   Start 2: SumTest.Zero
10  2/16 Test #2: SumTest.Zero ..... Passed    0.00 sec
11  Start 3: SumTest.Negative
12  3/16 Test #3: SumTest.Negative ..... Passed    0.00 sec
13  Start 4: SumTest.Big
14  4/16 Test #4: SumTest.Big ..... Passed    0.00 sec
15  Start 5: DiffTest.Basic
16  5/16 Test #5: DiffTest.Basic ..... Passed    0.00 sec
17  Start 6: DiffTest.Zero
18  6/16 Test #6: DiffTest.Zero ..... Passed    0.00 sec
19  Start 7: DiffTest.Negative
20  7/16 Test #7: DiffTest.Negative ..... Passed    0.00 sec
21  Start 8: DiffTest.Big
22  8/16 Test #8: DiffTest.Big ..... Passed    0.00 sec
23  Start 9: MultiTest.Basic
24  9/16 Test #9: MultiTest.Basic ..... Passed    0.00 sec
25  Start 10: MultiTest.Zero
26 10/16 Test #10: MultiTest.Zero ..... Passed    0.00 sec
27  Start 11: MultiTest.Negative
28 11/16 Test #11: MultiTest.Negative ..... Passed    0.00 sec
29  Start 12: MultiTest.Big
30 12/16 Test #12: MultiTest.Big ..... Passed    0.00 sec
31  Start 13: DivisionTest.Basic
32 13/16 Test #13: DivisionTest.Basic ..... Passed    0.00 sec
33  Start 14: DivisionTest.Zero
34 14/16 Test #14: DivisionTest.Zero ..... Passed    0.00 sec
35  Start 15: DivisionTest.Negative
36 15/16 Test #15: DivisionTest.Negative ..... Passed    0.00 sec
37  Start 16: DivisionTest.Big
38 16/16 Test #16: DivisionTest.Big ..... Passed    0.00 sec
39
40 100% tests passed, 0 tests failed out of 16
41
42 Total Test time (real) =  0.06 sec
```

- > Run marvinpinto/action-automatic-releases@latest
- > Post Run actions/checkout@v3
- > Complete job

5. Сделаем ошибку в файле и сделаем коммит.

```
13
14 TEST(SumTest, Negative) {
15     EXPECT_EQ(CuTlk( x: -6, y: 7, ch: '+'), 1);
16     EXPECT_EQ(CuTlk( x: -6, y: -10, ch: '+'), 16);
17 }
```

6. Запустится автоматическая сборка и мы получим ошибку.

```
build
failed 1 minute ago in 27s

> Set up job
> Run actions/checkout@v3
> Configure CMake
> Build
< Test
1 ▶ Run ctest -C Release
6 Test project /home/runner/work/NEW_YEAR_CULC2/NEW_YEAR_CULC2/build
7   Start 1: SumTest.Basic
8 1/16 Test #1: SumTest.Basic ..... Passed    0.00 sec
9   Start 2: SumTest.Zero
10 2/16 Test #2: SumTest.Zero ..... Passed    0.00 sec
11   Start 3: SumTest.Negative
12 3/16 Test #3: SumTest.Negative .....***Failed 0.00 sec
13   Start 4: SumTest.Big
14 4/16 Test #4: SumTest.Big ..... Passed    0.00 sec
15   Start 5: DiffTest.Basic
16 5/16 Test #5: DiffTest.Basic ..... Passed    0.00 sec
17   Start 6: DiffTest.Zero
18 6/16 Test #6: DiffTest.Zero ..... Passed    0.00 sec
19   Start 7: DiffTest.Negative
20 7/16 Test #7: DiffTest.Negative ..... Passed    0.00 sec
21   Start 8: DiffTest.Big
22 8/16 Test #8: DiffTest.Big ..... Passed    0.00 sec
23   Start 9: MultiTest.Basic
24 9/16 Test #9: MultiTest.Basic ..... Passed    0.00 sec
25   Start 10: MultiTest.Zero
26 10/16 Test #10: MultiTest.Zero ..... Passed    0.00 sec
27   Start 11: MultiTest.Negative
28 11/16 Test #11: MultiTest.Negative ..... Passed    0.00 sec
29   Start 12: MultiTest.Big
30 12/16 Test #12: MultiTest.Big ..... Passed    0.00 sec
31   Start 13: DivisionTest.Basic
32 13/16 Test #13: DivisionTest.Basic ..... Passed    0.00 sec
33   Start 14: DivisionTest.Zero
34 14/16 Test #14: DivisionTest.Zero ..... Passed    0.00 sec
35   Start 15: DivisionTest.Negative
36 15/16 Test #15: DivisionTest.Negative ..... Passed    0.00 sec
37   Start 16: DivisionTest.Big
38 16/16 Test #16: DivisionTest.Big ..... Passed    0.00 sec
39
40 94% tests passed, 1 tests failed out of 16
41
42 Total Test time (real) = 0.14 sec
43
44 The following tests FAILED:
45   3 - SumTest.Negative (Failed)
46 Errors while running CTest
47 Output from these tests are in: /home/runner/work/NEW_YEAR_CULC2/NEW_YEAR_CULC2/build/Testing/Temporary/LastTest.log
48 Use "--rerun-failed --output-on-failure" to re-run the failed cases verbosely.
49 Error: Process completed with exit code 8.

○ Run marvinpinto/action-automatic-releases@latest
> Post Run actions/checkout@v3
> Complete job
```

7. Организуем сборку в исполняемый файл. Добавим к нашему Yaml файлу следующие строки, в которых создаем release с

соответствующим тегом, прикладываем к нему исполняемый файл. Делаем коммит.

```
- uses: "marvinpinto/action-automatic-releases@latest"
  with:
    repo_token: "${{ secrets.GITHUB_TOKEN }}"
    automatic_release_tag: "latest"
    prerelease: true
    title: "New Year Build"
    files: |
      ${{github.workspace}}/build/culc_test
```

8. После этого коммита началась автоматическая сборка. Так как сейчас в файле находится ошибка, то исполняемый файл не соберется. Исправим ее, делаем коммит и увидим, что в репозитории появился новый тег, перейдя по которому увидим

