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 №M3103

Кравченкова Елизавета Александровна

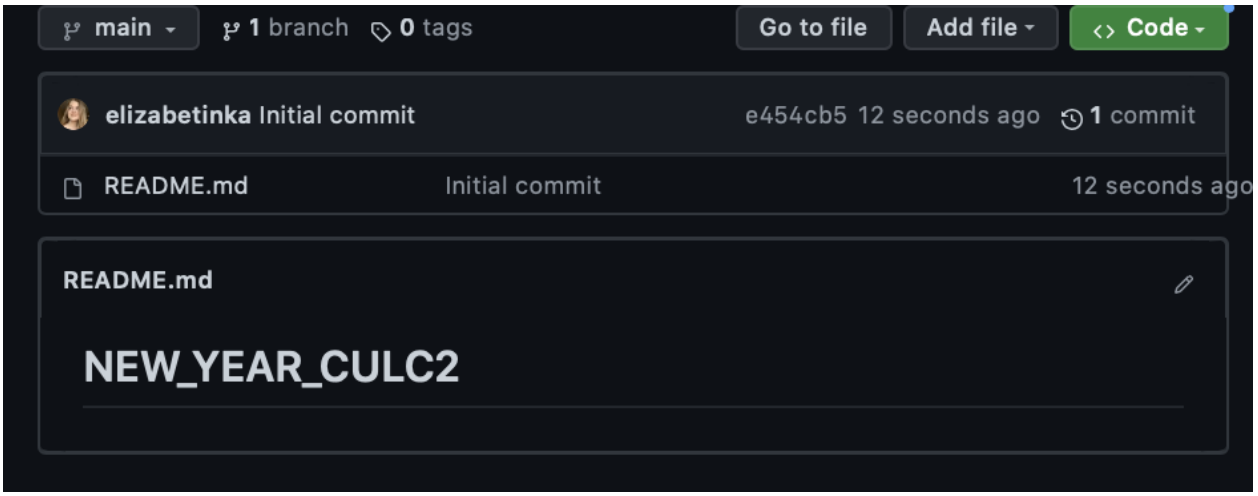
Проверил:

Повышев Владислав Вячеславович

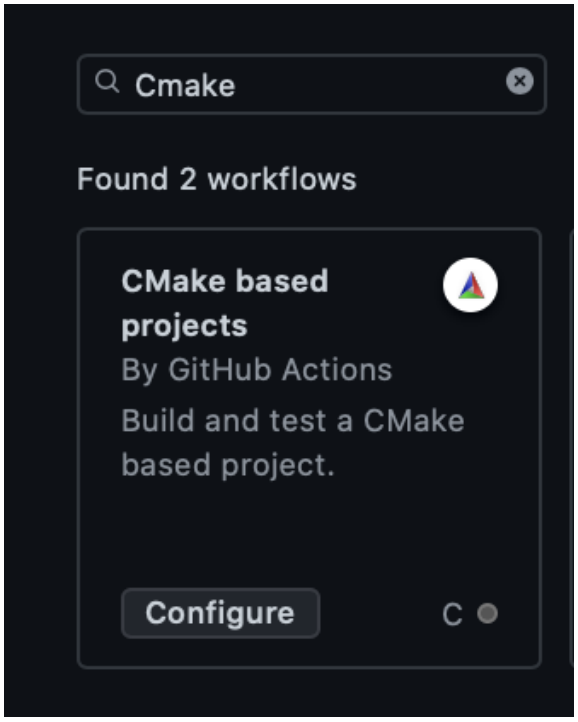
Санкт-Петербург

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

```yaml
name: CMake

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

env:
  # Customize the CMake build type here (Release, Debug, RelWithDebInfo, etc.)
  BUILD_TYPE: Release

jobs:
  build:
    # The CMake configure and build commands are platform agnostic and should work equally well on Windows or Mac.
    # You can convert this to a matrix build if you need cross-platform coverage.
    # See: https://docs.github.com/en/free-pro-team@latest/actions/learn-github-actions/managing-complex-workflows#using-a-build-matrix
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3

    - name: Configure CMake
      # Configure CMake in a 'build' subdirectory. `CMAKE_BUILD_TYPE` is only required if you are using a single-configuration generator such as make.
      # See https://cmake.org/cmake/help/latest/variable/CMAKE_BUILD_TYPE.html?highlight=cmake_build_type
      run: cmake -B ${{github.workspace}}/build -DCMAKE_BUILD_TYPE=${{env.BUILD_TYPE}}

    - name: Build
      # Build your program with the given configuration
      run: cmake --build ${{github.workspace}}/build --config ${{env.BUILD_TYPE}}

    - name: Test
      working-directory: ${{github.workspace}}/build
      # Execute tests defined by the CMake configuration.
      # See https://cmake.org/cmake/help/latest/manual/ctest.1.html for more detail
      run: ctest -C ${{env.BUILD_TYPE}}
```

Cancel changes | Start commit

Marketplace | Documentation

Search Marketplace for Actions

Featured Actions

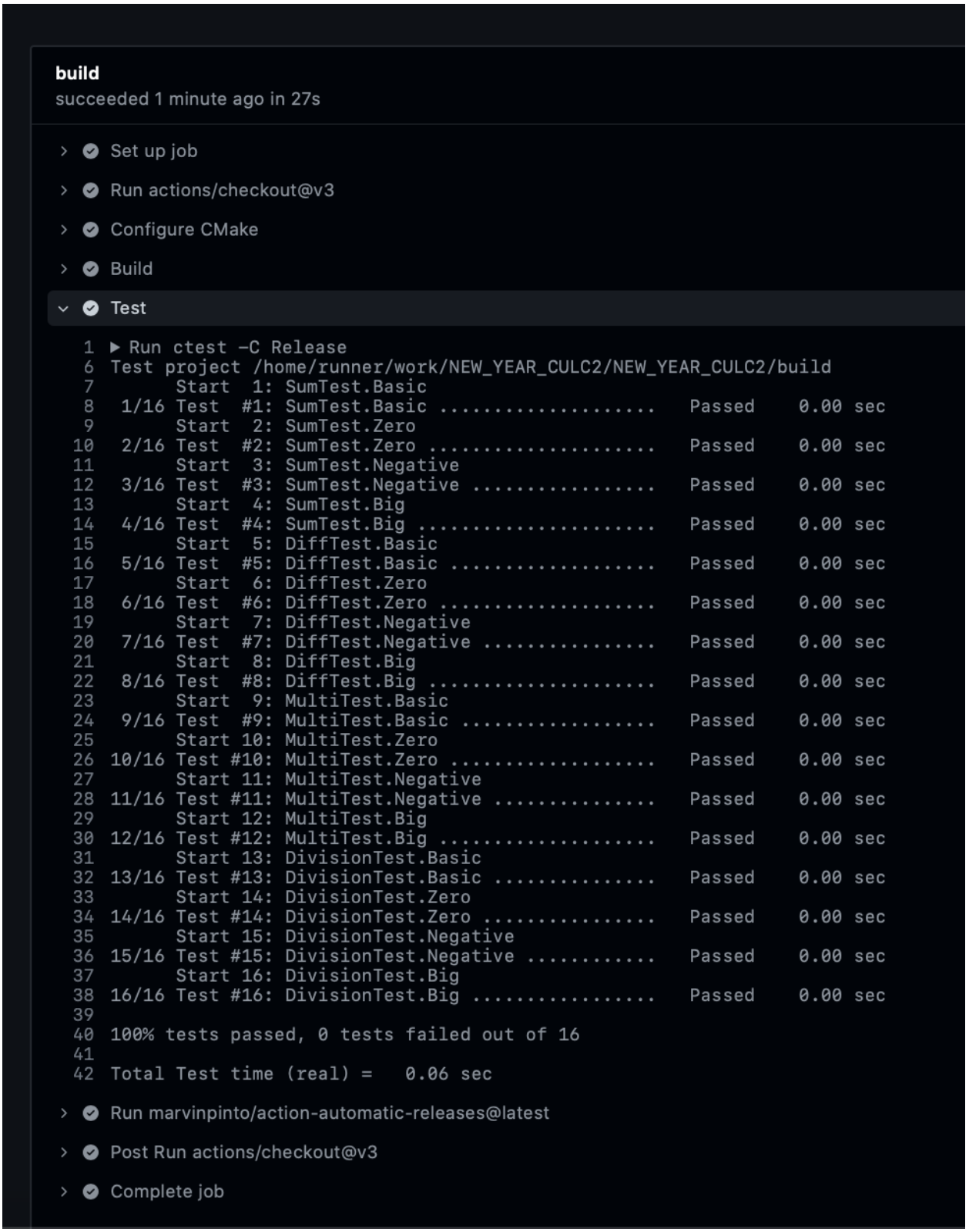
Cache ☆ 3.3k
By actions
Cache artifacts like dependencies and build outputs to improve workflow execution time

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

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

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

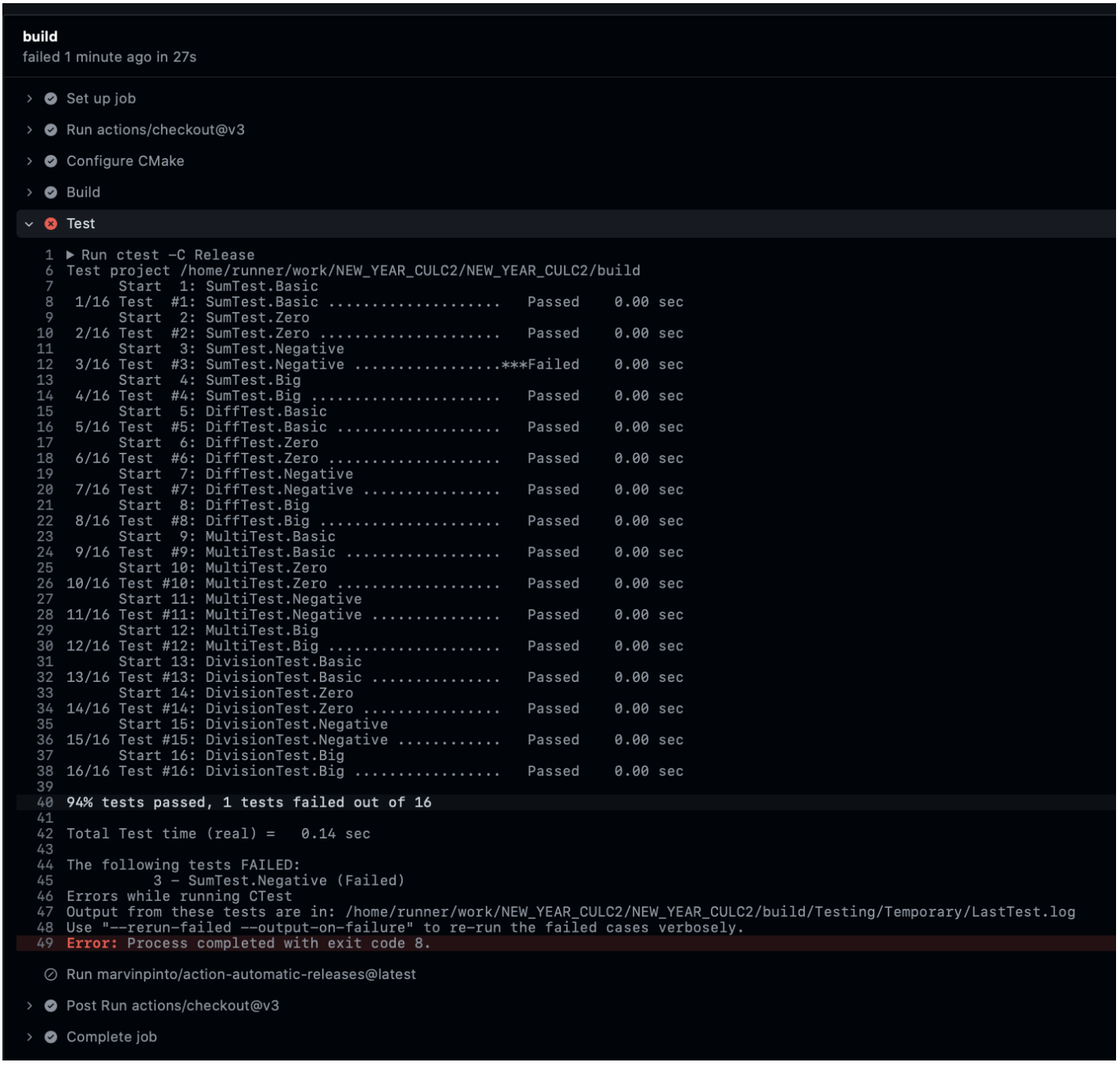
4. После коммита зайдем в раздел Actions и там увидем, что началась автоматическая сборка проекта. Заметим, что сборка прошла успешно, все тесты пройдены



5. Сделаем ошибку в файле и сделаем коммит.

```
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  }
```

6. Запустится автоматическая сборка и мы получим ошибку.



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.

```yaml
- 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 увидем