



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)  
КАФЕДРА «Информационная безопасность» (ИУ8)

Лабораторная работа № 5  
ПО КУРСУ  
«Технологии и методы программирования»  
на тему «Изучение систем контроля версий на примере Git»

Студент

ИУ8-24  
(Группа)

Т. И. Андронов  
(И. О. Фамилия)

Преподаватель:

А. А. Кодык  
(И.О. Фамилия)

## Ход работы:

1. Создайте CMakeList.txt для библиотеки *banking*.
2. Создайте модульные тесты на классы Transaction и Account.
  - Используйте mock-объекты.
  - Покрытие кода должно составлять 100%.
3. Настройте сборочную процедуру на **TravisCI**.
4. Настройте [Coveralls.io](https://coveralls.io).

```
1 cmake_minimum_required(VERSION 3.4)
2 set(CMAKE_CXX_STANDARD 11)
3 set(CMAKE_CXX_STANDARD_REQUIRED ON)
4 project(task1)
5
6 add_library(banking STATIC ${CMAKE_CURRENT_SOURCE_DIR}/Account.cpp ${CMAKE_CURRENT_SOURCE_DIR}/Transaction.cpp)
```

```
1 cmake_minimum_required(VERSION 3.4)
2 set(CMAKE_CXX_STANDARD 11)
3 set(CMAKE_CXX_STANDARD_REQUIRED ON)
4 option(BUILD_TESTS "Build tests" OFF)
5 project(test)
6
7 set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} --coverage")
8
9 include_directories(PUBLIC ${CMAKE_CURRENT_SOURCE_DIR}/banking)
10 add_library(banking STATIC ${CMAKE_CURRENT_SOURCE_DIR}/../banking/Transaction.cpp
11                               ${CMAKE_CURRENT_SOURCE_DIR}/../banking/Account.cpp)
12
13 if(BUILD_TESTS)
14     enable_testing()
15     add_subdirectory(third-party/gtest)
16     add_executable(check test1.cpp test2.cpp)
17     target_link_libraries(check banking gtest_main)
18     add_test(NAME check COMMAND check)
19 endif()
```

---

```
1  #include "gtest/gtest.h"
2  #include "../banking/Account.h"
3
4  TEST(Acc, testall)
5  {
6      Account acc(12312, 370);
7
8      //check get
9      EXPECT_EQ(acc.GetBalance(), 370);
10
11     //check lock/unlock
12     EXPECT_THROW(acc.ChangeBalance(70), std::runtime_error);
13     acc.Lock();
14     EXPECT_NO_THROW(acc.ChangeBalance(30));
15     EXPECT_EQ(acc.GetBalance(), 400);
16     EXPECT_THROW(acc.Lock(), std::runtime_error);
17     acc.Unlock();
18     EXPECT_THROW(acc.ChangeBalance(37), std::runtime_error);
19 }
20
```

---

```
1  #include "../banking/Account.h"
2  #include "../banking/Transaction.h"
3  #include "gtest/gtest.h"
4
5  TEST(Transaction, Banking)
6  {
7      const int sum1 = 500, sum2 = 300, fee = 100;
8
9      Account a1(37, sum1), a2(73, sum2);
10     Transaction test;
11
12     //set & get
13     EXPECT_EQ(test.fee(), 1);
14     test.set_fee(fee);
15     EXPECT_EQ(test.fee(), fee);
16
17     //check exceptions
18     EXPECT_THROW(test.Make(a1, a1, 100), std::logic_error);
19     EXPECT_THROW(test.Make(a1, a2, -10), std::invalid_argument);
20     EXPECT_THROW(test.Make(a1, a2, 10), std::logic_error);
21     EXPECT_EQ(test.Make(a1, a2, test.fee()*2-1), false);
22     a2.Lock();
23     EXPECT_THROW(test.Make(a1, a2, 200), std::runtime_error);
24     a2.Unlock();
25
26     //check cancelled transaction
27     EXPECT_EQ(test.Make(a1, a2, sum1-fee), false);
28     EXPECT_EQ(a2.GetBalance(), sum2);
29     EXPECT_EQ(a1.GetBalance(), sum1);
30
31     //check successful transaction
32     EXPECT_EQ(test.Make(a1, a2, 300), true);
33     EXPECT_EQ(a2.GetBalance(), sum2+300);
34     EXPECT_EQ(a1.GetBalance(), sum1-300-fee);
35 }
```

## Workflow file for this run

.github/workflows/main.yml at 404b9a4

```
1  name: testing
2  on: [push]
3
4  jobs:
5    build:
6      runs-on: ubuntu-latest
7      steps:
8        - name: checkout
9          uses: actions/checkout@v3
10
11        - name: install lcov
12          run: sudo apt-get install -y lcov
13
14        - name: update_gtest
15          run: |
16            cd tests/third-party/gtest
17            git submodule init && git submodule update
18
19        - name: build_tests
20          shell: bash
21          run: |
22            cd tests/
23            cmake -H. -B_build -D BUILD_TESTS=ON
24            cmake --build _build
25
26        - name: run_tests
27          shell: bash
28          run: tests/_build/check
29
30        - name: creating_lcov_file
31          run: lcov -o ./coverage/lcov.info -c -d ./tests/_build/CMakeFiles/check.dir/ --include *.cpp
32
33        - name: coveralls
34          uses: coverallsapp/github-action@master
35          with:
36            github-token: ${ secrets.GITHUB_TOKEN }
```

REPO ADDED 16 MAY 2022 11:36AM UTC	TOTAL FILES 2	# BUILDS 1	BADGE coverage 100%	TOKEN Qu61d742UuBePvtWxvIXkr5gkzcoE
---------------------------------------	------------------	---------------	------------------------	--

## LAST BUILD ON BRANCH MASTER

P BRANCH: MASTER

COMMITTED 16 MAY 2022 - 11:52

FIRST BUILD ON MASTER AT 100.0%

BUILD #	BUILD TYPE	COMMITTED BY	COMMIT MESSAGE	RUN DETAILS
2331699619	push github	GitHub	test2	32 of 32 relevant lines covered (100.0%) 1.31 hits per line

## RELEVANT LINES COVERED

## SOURCE FILES ON MASTER

TREE LIST 2 CHANGED 0 SOURCE CHANGED 0 COVERAGE CHANGED 0

100.0 tests/

## RECENT BUILDS

BUILDS	BRANCH	COVERAGE	COMMIT	COMMITTER	TYPE	TIME	VIA
2331699619	master	100.0	test2	GitHub	push	16 May 2022 11:52AM UTC	github