

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

**Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики**
Факультет Фотоники и оптоинформатики
Кафедра Компьютерной фотоники и видеоинформатики

Отчет по практике

Выполнила:
Старобыховская А. А.

Группа: V3316

Преподаватель: Кудрявцев А. С.

Санкт-Петербург, 2017

Оглавление

Цель проведения практики	3
Задание №1. Знакомство с системой контроля версий Git	4
Задание №2	12
Приложение А. Слайды презентации с семинара по C++11 на тему:	18

Цель проведения практики

Освоение навыков использования C++ и изучение приемов разработки программного обеспечения. Практика проходит в компьютерном классе и состоит из лекционных занятий и практических заданий.

Задание №1. Знакомство с системой контроля версий Git

1. Завести аккаунт на github.com. На рис.1 скриншот с сайта github.com.

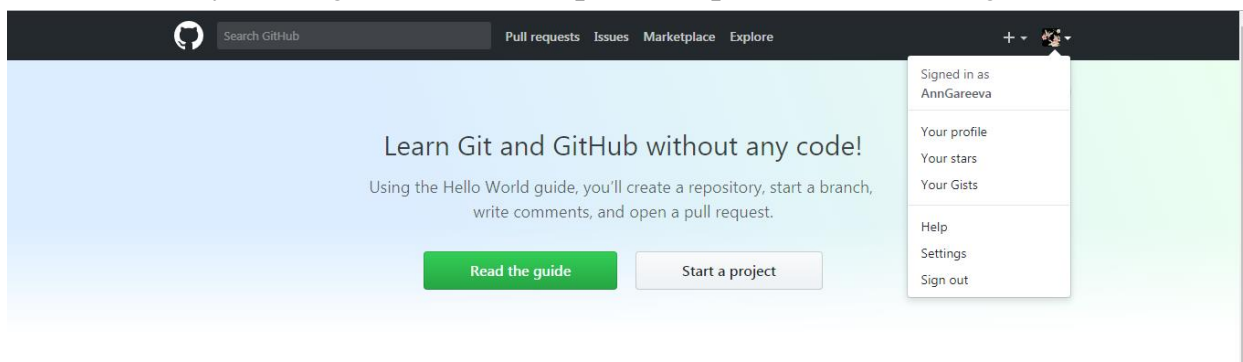


Рисунок 1- создание своего аккаунта на сайте github.com

2. Создать репозиторий для лабораторных работ. Рис.2 демонстрирует процесс создания репозитория, а рис.3 факт наличия его.

A screenshot of the "Create a new repository" form on GitHub. The form title is "Create a new repository" with a subtitle "A repository contains all the files for your project, including the revision history." The form fields include: "Owner" (a dropdown menu showing "AnnGareeva"), "Repository name" (a text input field containing "PracticeAAS" with a green checkmark), "Description (optional)" (a text input field containing "Lab of practice 2017"), "Visibility" (radio buttons for "Public" (selected) and "Private"), "Initialize this repository with a README" (checkbox), "Add .gitignore" (a dropdown menu showing "None"), and "Add a license" (a dropdown menu showing "None" with an information icon). A green "Create repository" button is at the bottom.

Рисунок 2-создание репозитория на сайте

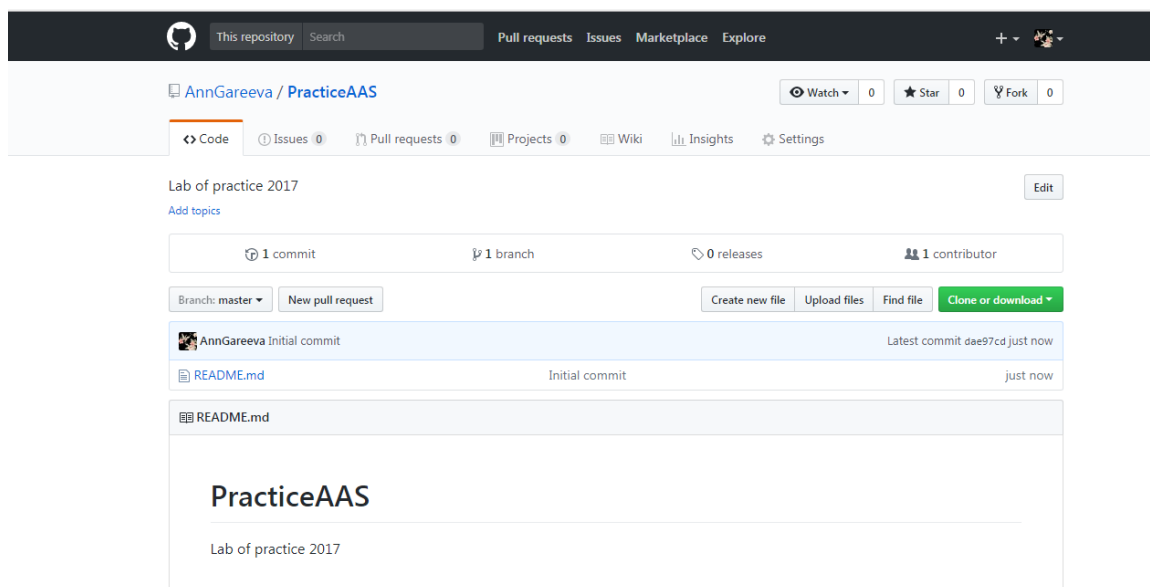


Рисунок 3-создание репозитория на сайте

3. Склонировать репозиторий на компьютер.

Для того, чтобы клонировать созданный репозиторий, необходимо скопировать на него ссылку. Именно это демонстрирует рис. 4.

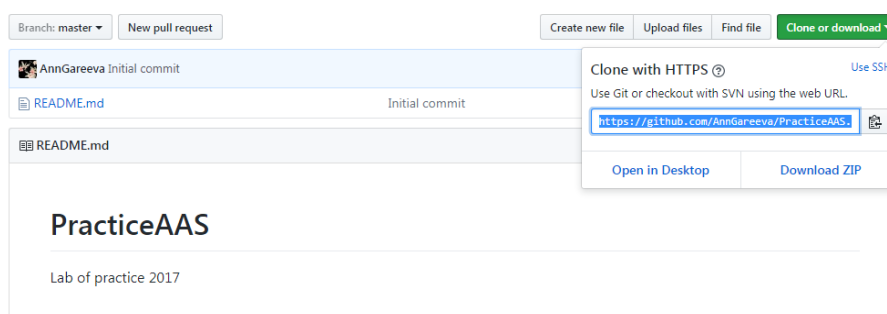
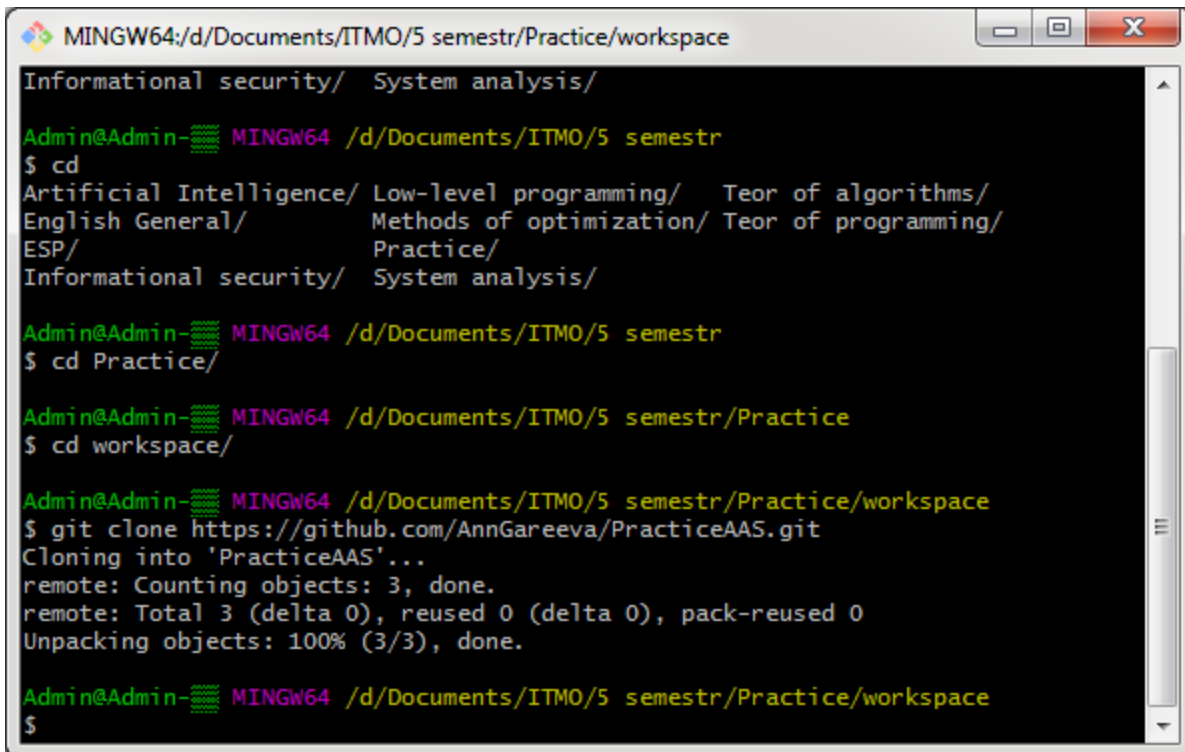


Рисунок 4-копирование ссылки на репозиторий для его клонирования

Скриншот Git-Bush показывает результаты выполнения команды `git clone` [<ссылка>](#) (рис. 5)



```

MINGW64:/d/Documents/ITMO/5 semestr/Practice/workspace
Informational security/ System analysis/

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr
$ cd
Artificial Intelligence/ Low-level programming/ Teor of algorithms/
English General/ Methods of optimization/ Teor of programming/
ESP/ Practice/
Informational security/ System analysis/

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr
$ cd Practice/

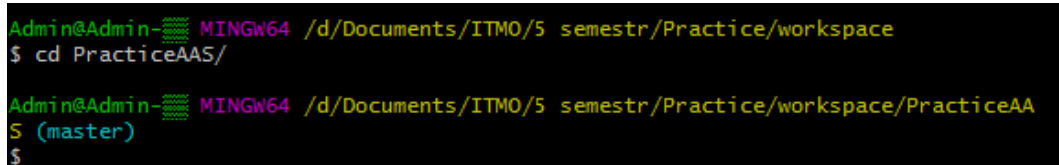
Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice
$ cd workspace/

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace
$ git clone https://github.com/AnnGareeva/PracticeAAS.git
Cloning into 'PracticeAAS'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace
$
  
```

Рисунок 5-результат выполнения команды `git clone`

После перехода в клонированную папку появляется `(master)`, дающий нам понять, что мы находимся и работаем в главной ветке. Это показано на рис.6.



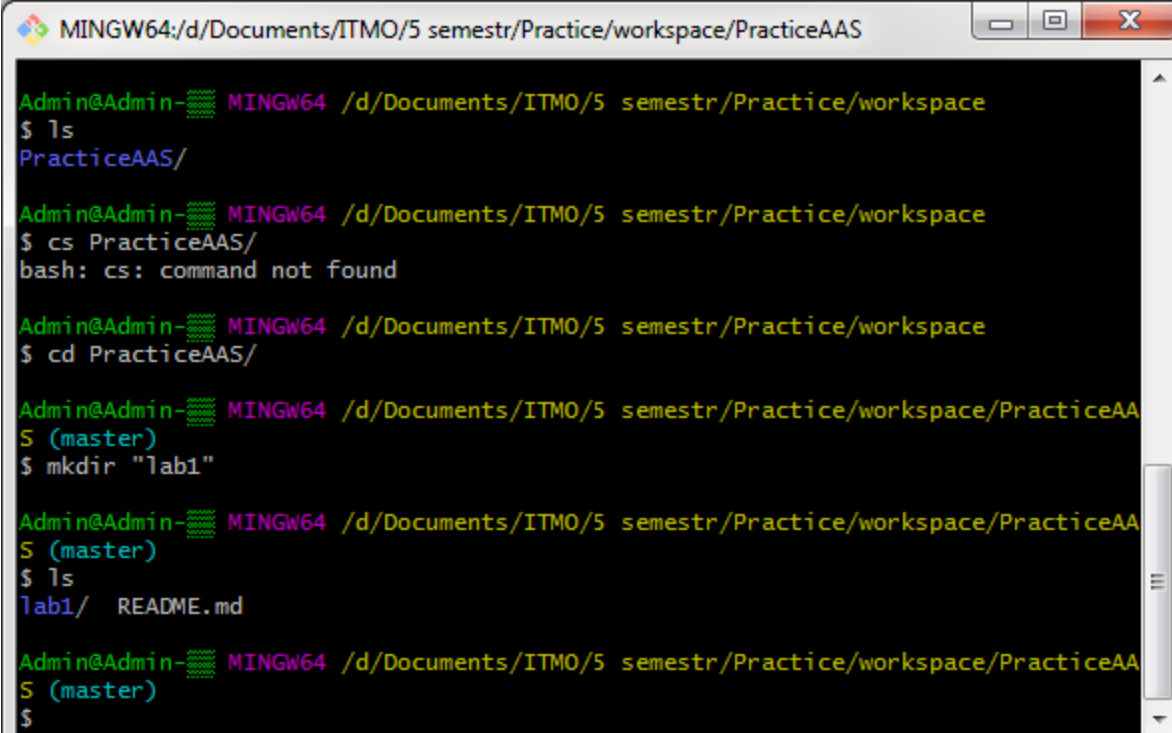
```

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace
$ cd PracticeAAS/

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
$ (master)
$
  
```

Рисунок 6- ветка master

4. Создать папку lab1. С помощью команды mkdir “название”, создаем папку (рис.7) и переходим в нее.



```

MINGW64:/d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAAS
Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace
$ ls
PracticeAAS/

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace
$ cs PracticeAAS/
bash: cs: command not found

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace
$ cd PracticeAAS/

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$ mkdir "lab1"

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$ ls
lab1/  README.md

Admin@Admin-MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$
  
```

Рисунок 7-создание папки для выполнения л.р. 1

5. Написать в папке lab1 программу, вычисляющую функцию факториала.

Код программы:

```

int Factorial(int a) {
    return (a == 1) ? 1 : Factorial(a - 1)*a;
}
  
```

6. Создать .gitignore, добавив в игнорируемые файлы – промежуточные файлы компиляции (объектные файлы и другие временные файлы) и исполняемый файл. В результате должны остаться только файл проекта и исходные файлы.


```

new file:   .gitignore
new file:   lab1/Pract_l1_AAS/.vs/Pract_l1_AAS/v15/.suo
new file:   lab1/Pract_l1_AAS/.vs/Pract_l1_AAS/v15/Browse.VC.db
new file:   lab1/Pract_l1_AAS/.vs/Pract_l1_AAS/v15/ipch/AutoPCH/7576442e
121da2ab/SOURCE.ipch
new file:   lab1/Pract_l1_AAS/Pract_l1_AAS/Source.cpp

Admin@Admin- MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$ git commit
[master 38ad1ff] Created first version of prog (factorial).
5 files changed, 15 insertions(+)
create mode 100644 .gitignore
create mode 100644 lab1/Pract_l1_AAS/.vs/Pract_l1_AAS/v15/.suo
create mode 100644 lab1/Pract_l1_AAS/.vs/Pract_l1_AAS/v15/Browse.VC.db
create mode 100644 lab1/Pract_l1_AAS/.vs/Pract_l1_AAS/v15/ipch/AutoPCH/7576442e
121da2ab/SOURCE.ipch
create mode 100644 lab1/Pract_l1_AAS/Pract_l1_AAS/Source.cpp

Admin@Admin- MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$

```

Рисунок 10-результат выполнения команды git commit

Создание второго коммита после изменения файла. Новый код:

```

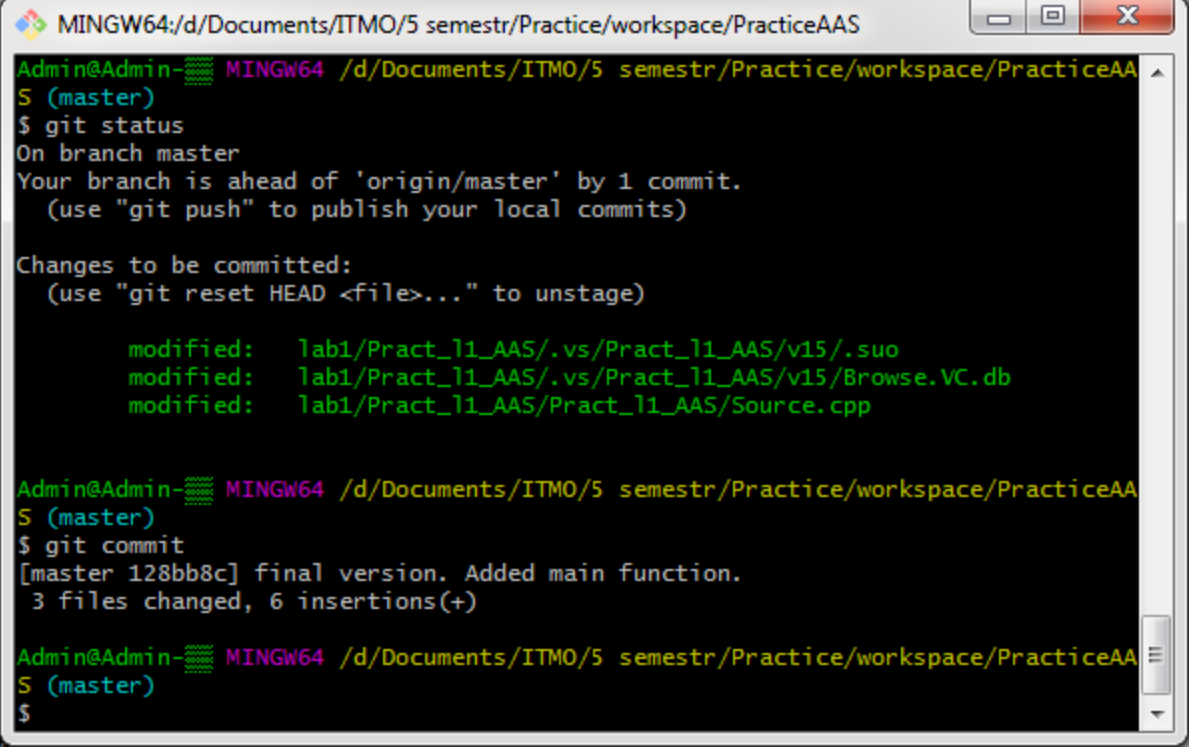
#include <iostream>
int Factorial(int a) {
    return (a == 1) ? 1 : Factorial(a - 1)*a;
}

#Function testing

int main() {
    int x = 12;
    std::cout << Factorial(x) << std::endl;
}

```

Результат выполнения команды `git commit` показан на рис. 11.



```

MINGW64:/d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAAS
Admin@Admin- MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

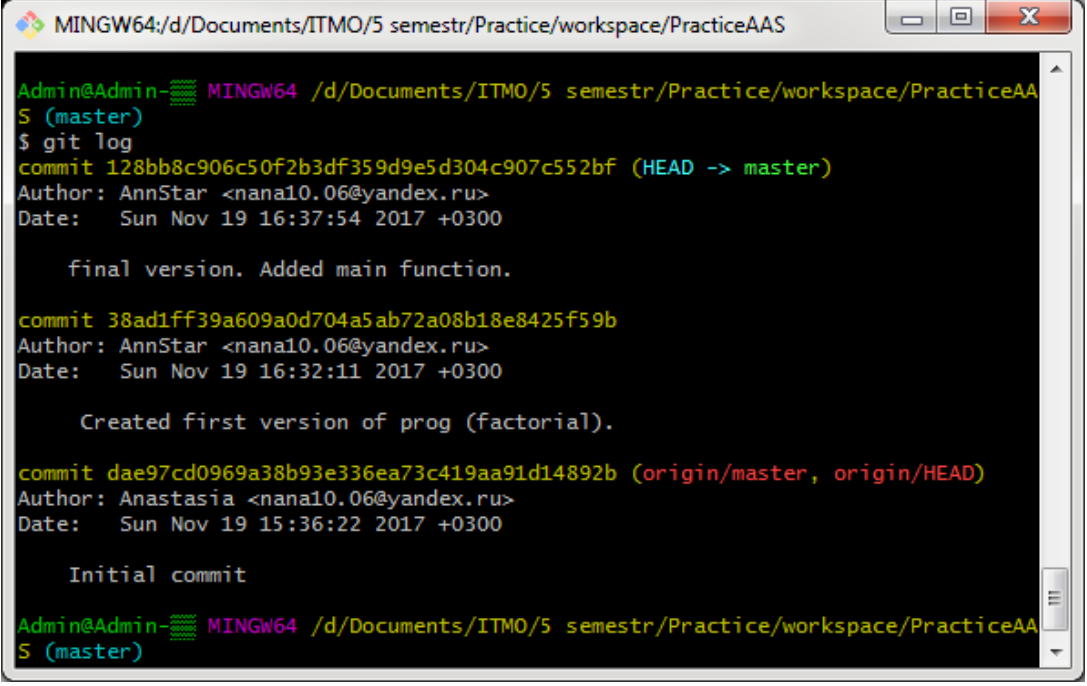
        modified:   lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/.suo
        modified:   lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/Browse.VC.db
        modified:   lab1/Pract_11_AAS/Pract_11_AAS/Source.cpp

Admin@Admin- MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$ git commit
[master 128bb8c] final version. Added main function.
 3 files changed, 6 insertions(+)

Admin@Admin- MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$
  
```

Рисунок 11-результат выполнений команды `git commit` после изменения файла

8. Привести результаты работы “`git log`” (рис. 12).



```

MINGW64:/d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAAS
Admin@Admin- MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$ git log
commit 128bb8c906c50f2b3df359d9e5d304c907c552bf (HEAD -> master)
Author: AnnStar <nana10.06@yandex.ru>
Date:   Sun Nov 19 16:37:54 2017 +0300

    final version. Added main function.

commit 38ad1ff39a609a0d704a5ab72a08b18e8425f59b
Author: AnnStar <nana10.06@yandex.ru>
Date:   Sun Nov 19 16:32:11 2017 +0300

    Created first version of prog (factorial).

commit dae97cd0969a38b93e336ea73c419aa91d14892b (origin/master, origin/HEAD)
Author: Anastasia <nana10.06@yandex.ru>
Date:   Sun Nov 19 15:36:22 2017 +0300

    Initial commit

Admin@Admin- MINGW64 /d/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$
  
```

Рисунок 12-результат выполнения команды `git log`

9. Показать при помощи “git diff” изменение любого файла между двумя коммитами. Результат выполнения данной команды показан на рис.13.

```

MINGW64;d:/Documents/ITMO/5 semestr/Practice/workspace/PracticeAAS
$ cd "D:\Documents\ITMO\5 semestr\Practice\workspace\PracticeAAS"

Admin@Admin-MINGW64 /d:/Documents/ITMO/5 semestr/Practice/workspace/PracticeAA
S (master)
$ git diff 38ad1ff39a609a0d704a5ab72a08b18e8425f59b dae97cd0969a38b93e336ea73c419aa91d14892b
diff --git a/.gitignore b/.gitignore
deleted file mode 100644
index a3c2731..0000000
--- a/.gitignore
+++ /dev/null
@@ -1,12 +0,0 @@
-*.sln
-*.exe
-*.ilk
-*.pdb
-*.vcxproj
-*.vcxproj.filters
-*.log
-*.obj
-*.idb
-*.pdb
-*.tlog
-*.lastbuildstate
diff --git a/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/.suo b/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/.suo
deleted file mode 100644
index 62cbb2a..0000000
Binary files a/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/.suo and /dev/null differ
diff --git a/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/Browse.VC.db b/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/Browse.VC.db
deleted file mode 100644
index 61e7175..0000000
Binary files a/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/Browse.VC.db and /dev/null differ
diff --git a/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/ipch/AutoPCH/7576442e121da2ab/SOURCE.ipch b/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/ipch/AutoPCH/7576442e121da2ab/SOURCE.ipch
deleted file mode 100644
index 87ccc54..0000000
Binary files a/lab1/Pract_11_AAS/.vs/Pract_11_AAS/v15/ipch/AutoPCH/7576442e121da2ab/SOURCE.ipch and /dev/null differ
diff --git a/lab1/Pract_11_AAS/Pract_11_AAS/Source.cpp b/lab1/Pract_11_AAS/Pract_11_AAS/Source.cpp
deleted file mode 100644
index 34e7837..0000000
--- a/lab1/Pract_11_AAS/Pract_11_AAS/Source.cpp
+++ /dev/null
@@ -1,3 +0,0 @@
-int Factorial(int a) {
-    return (a == 1) ? 1 : Factorial(a - 1)*a;
-}

```

Рисунок 13-результат выполнения команды git diff

Задание №2. Форматирование и стиль.

Условия задания

Необходимо реализовать задачу №1079 «Максимум».

Ограничение времени: 2.0 секунды

Ограничение памяти: 64 МБ

Рассмотрим последовательность чисел a_i , $i = 0, 1, 2, \dots$, удовлетворяющих следующим условиям:

- $a_0 = 0$
- $a_1 = 1$
- $a_{2i} = a_i$
- $a_{2i+1} = a_i + a_{i+1}$,

где для каждого $i = 1, 2, 3, \dots$.

Цель: написать программу, которая для заданного значения n находит максимальное среди чисел a_0, a_1, \dots, a_n .

Исходные данные: входные данные состоят из нескольких тестов (не более 10). Каждый тест представляет собой строку, в которой записано целое число n ($1 \leq n \leq 99\,999$). В последней строке входных данных записано число 0.

Результат: для каждого n во вводе выведите соответствующее максимальное значение.

Пример

Исходные данные	Результат
5	3
10	4
0	
10000	512
10000	512
0	
0	

Алгоритм решения

В первую очередь, необходимо считать и сохранить все введенные данные. Считываем данные до тех пор, пока введенное значение не равно 0. Если значение равно 0, то считывание завершается. Создаем вектор, в котором будет храниться ряд значений a . В данный вектор помещаем первые два элемента, равные 0 и 1. Считываем введенный элемент и заходим в цикл от 2 до значения элемента с шагом 1. Проверяем на четность значение элемента. Если значение четное, то в вектор ряда складываем следующий элемент, который будет равен элементу этого ряда с индексом, который равен половине текущего итератора в цикле. Если значение нечетно, то значение следующего элемента будет равно сумме элементов, индексы которых равны половине текущего и половине текущего плюс 1 соответственно. После того, как получили ряд для одного введенного числа, ищем максимальный его элемент и выводим. Аналогично поступаем для всех введенных тестовых данных.

Текст программы

```
#include <iostream>
#include <vector>

using namespace std;

int main() {
    vector<long long> values;
    vector<long long> test;
    long long num = -1;
    if (num == 0) {
        return 1;
    }
    cin >> num;
    while (num != 0) {
        test.push_back(num);
        cin >> num;
    }
    system("cls");
    for (long long i = 0; i < test.size(); i++) {
        values.push_back(0);
        values.push_back(1);
        for (long long j = 2; j < (test[i] + 1); j++) {
            long long index = 0;
            if (j % 2 == 0) {
                index = j / 2;
                values.push_back(values[index]);
            }
            else {
                index = (j - 1) / 2;
                values.push_back(values[index] + values[index + 1]);
            }
        }
        long long maximum = 0;
        for (int j = 0; j < values.size(); j++) {
            if (maximum < values[j]) maximum = values[j];
        }
        cout << maximum;
        if (i != (test.size() - 1)) {
            cout << endl;
        }
        values.clear();
    }
    system("pause");
    return 0;
}
```

Результат прохождения тестов на сайте acm.timus.ru представлен на рисунке 14.

Результаты проверки решений

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
7653564	13:09:15 4 дек 2017 13:05:25	AAS	1079_Максимум	Visual C++ 2017	Accepted		0.031	1 740 КБ

Рисунок 14- результаты прохождения тестов программой.

Для 2й и последующих л/р:

1. Условия задания.
2. Схема алгоритма решения задачи (словесное описание пути решения и/или математическая модель задачи (формулы, алгоритмы ...)).
3. Текст программы.
4. Исходные данные, используемые при вводе в программу.
5. Результаты выполнения программы.

Дополнительные указания

1. Условие задания с сайта [timus](#) приводить полностью (включая номер и название задания, описание, исходные данные, результат и пример), лучше помещать задание в отчет как рисунок;
2. Помещать в отчет скриншот сайта [timus](#) с результатами проверки, включающий логин автора, номер задания и результат выполнения;
3. Предоставлять для проверки отчет, сохраненный в формате pdf (т.к. в разных версиях MS Word документ может отображаться по-разному, внося искажения в оформление отчета, что снижает оценку за оформление).
4. Бумажная версия отчета предоставляется в конце семестра. Оформление отчета по ГОСТ 7.32-200х.

Основные требования и рекомендации по оформлению (составлено по наиболее частым ошибкам)

1. Отчет должен иметь титульный лист, оглавление и содержать отчеты по лабораторным работам, а также слайды презентации.
2. Страницы должны быть пронумерованы.

3. Для заголовков, подзаголовков, абзацев основного текста, списков и подписей рисунков/таблиц рекомендуется использовать стили. Это позволит обеспечить единообразное оформление и возможность быстрой корректировки оформления. Список стилей:

- Для заголовков: «Заголовок1», «Заголовок2», «Заголовок3».
- Для текста абзаца: «Обычный».
- Для рисунков и подписей: «Рисунок».
- Для списков: «Список-номер», «Список-тире».

4. Рисунки и таблицы должны иметь номера и названия. Не допускается расположение рисунка/таблицы и его подписи на разных страницах. Формулы должны быть пронумерованы. На рисунок и таблицу в тексте должны быть сноски. Примеры: (см. рис.1) (см. таблицу 1).

5. Презентацию доклада оформить в отчете как приложение.

Приложение А. Слайды презентации с семинара по C++11
на тему: