МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по курсовой работе

по дисциплине «Программирование»

Тема: Генерация отчетов

Студент гр. 7382 _____ Глазунов С.А.

Преподаватель _____ Кринкин К.В.

Санкт-Петербург

2018

ЗАДАНИЕ

НА КУРСОВУЮ РАБОТУ

Студент Глазунов С.А.

Группа 7382

Тема работы : Генерация отчетов

Исходные данные: В качестве основы для курсовой работы используется код лабораторной работы No4.

Содержание пояснительной записки:«Введение», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 5 страниц.

Дата выдачи задания: 28.11.2017

Дата сдачи реферата: 23.12.2017

Дата защиты реферата: 23.12.2017

Студент _____ Глазунов С.А.

Преподаватель _____ Кринкин К.В.

Для работы программы вам необходимо иметь

• Репозитой, в котором у вас есть comitt-доступ.
• *wiki*, можно брать *wiki* даже с не вашего репозитория.

Перед запуском вам необходимо настроить конфигурационный файл(*settings.json*)

Краткая настройка *settings.json*

Основные поля:
• *student* - ФИО студента.
• *teacher* - ФИО преподавателя.
• *download* - список имен файлов, которые будут добавлены в отчет в приложение.
• *PDF* - *bool* переменная, отвечающая за формат файла(*.pdf - True, .docx - False*).
• *pagesofwiki* - список страниц из *wiki*, которые надо включить в отчет.

Пример как протестировать данную программу

1. Если нет репозитория, в котором вы можете сделать *comitt*, то можете сделать *fork* [репозитория](https://github.com/light5551/test*gen)*

*2. Запустите программу как указано [здесь] (https://github.com/OSLL/report*generator#report*generator)*

*> Для тестирования wiki можно брать [отсюда] (https://github.com/light5551/test*gen/wiki), также для примера там будет ветка **test*branch***

Для работы программы вам необходимо иметь

Репозитой, в котором у вас есть comitt-доступ.
• *wiki*, можно брать *wiki* даже с не вашего репозитория.

Перед запуском вам необходимо настроить конфигурационный файл(*settings.json*)

Краткая настройка *settings.json*

Основные поля:
• *student* - ФИО студента.

• *teacher* - ФИО преподавателя.
• *download* - список имен файлов, которые будут добавлены в отчет в приложение.
• *PDF* - *bool* переменная, отвечающая за формат файла(*.pdf - True, .docx - False*).
• *pagesofwiki* - список страниц из *wiki*, которые надо включить в отчет.

Пример как протестировать данную программу

1. Если нет репозитория, в котором вы можете сделать *comitt*, то можете сделать *fork* [репозитория](https://github.com/light5551/test*gen)*

*2. Запустите программу как указано [здесь] (https://github.com/OSLL/report*generator#report*generator)*

*> Для тестирования wiki можно брать [отсюда] (https://github.com/light5551/test*gen/wiki), также для примера там будет ветка **test*branch***

Для работы программы вам необходимо иметь

Репозитой, в котором у вас есть comitt-доступ.
• *wiki*, можно брать *wiki* даже с не вашего репозитория.

Перед запуском вам необходимо настроить конфигурационный файл(*settings.json*)

Краткая настройка *settings.json*

Основные поля:
• *student* - ФИО студента.
• *teacher* - ФИО преподавателя.
• *download* - список имен файлов, которые будут добавлены в отчет в приложение.
• *PDF* - *bool* переменная, отвечающая за формат файла(*.pdf - True, .docx - False*).
• *pagesofwiki* - список страниц из *wiki*, которые надо включить в отчет.

Пример как протестировать данную программу

1. Если нет репозитория, в котором вы можете сделать *comitt*, то можете сделать *fork* [репозитория](https://github.com/light5551/test*gen)*

2. Запустите программу как указано [здесь]
*(https://github.com/OSLL/report*generator#report*generator)*

> Для тестирования wiki можно брать [отсюда]
*(https://github.com/light5551/test*gen/wiki), также для примера там будет ветка **test*branch***

Приложение

test.c

```
from random import randint

def bubble(array):
    for i in range(N-1):
        for j in range(N-i-1):
            if array[j] > array[j+1]:
                buff = array[j]
                array[j] = array[j+1]
                array[j+1] = buff

N = 10
a = []
for i in range(N):
    a.append(randint(1, 99))

print(a)
bubble(a)
print(a)
```

test2.c

```
//Следующая программа сортирует список чисел и выводит результат:
#include <stdio.h>
#include <stdlib.h>

int num[10] = {
1,3,6,5,8,7,9,6,2,0
};

int comp (const int *, const int *);

int main(void)
{
int i;
printf("Original array: ");
    for (i=0; i<10; i + +)
    {
        printf("%d ",num[i]);
    }
 printf ("\n");
 qsort(num, 10, sizeof (int), (int(*) (const void *, const void *)) comp);
 printf("Sorted array: ");
    for(i = 0; i <10; i + + )
        printf("%d ", num[i]);
return 0;
}

/* сравнение двух целых */
int comp (const int *i, const int *j)
{
  return *i - *j;
}
```