

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

гр. 7382

Преподаватель

Глазунов С.А.

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.

Аннотация. Необходимо, имея код лабораторной работы No4, реализовать алгоритм, который делает следующие преобразования со списком: Отсортировать список по невозрастанию по полю ueag в этом списке; Менять местами элементы не затрагивая поля, кроме тех, что указывают наследующие и предыдущие элементы; Пишется две функции, которые производят все эти преобразования и возвращают головной элемент списка

Аннотация. Необходимо, имея код лабораторной работы No4, реализовать алгоритм, который делает следующие преобразования со списком: Отсортировать список по невозрастанию по полю ueag в этом списке; Менять местами элементы не затрагивая поля, кроме тех, что указывают наследующие и предыдущие элементы; Пишется две функции, которые производят все эти преобразования и возвращают головной элемент списка

Введение. Необходимо, имея код лабораторной работы No4, реализовать алгоритм, который делает следующие преобразования со списком:

- Отсортировать список по невозрастанию по полю year в этом списке;
- Менять элементы не трогая поля, кроме тех, которые указывают на следующий и на предыдущий элемент

Welcome to the test_gen wiki!

Welcome to the test_gen wiki!

Приложение

main.py

```
1 #!/venv/bin/python3.6
2 import argparse
3 import sys
4 import os
5 import shutil
6 from mygithub import Gengit
7 from word import Dword
8
9
10 TIME_REPORT = "ready_project.docx"
11 READY_WORD = "generated_doc.docx"
12 FROM_CONSOLE = "cmd"
13 PDF = "PDF"
14 PDF_EXTENSION = "{}.pdf"
15 ERROR_MESSAGE = "Неверные входные данные!"
16 INPUT_URL_MESSAGE = "url of repo(ssh): "
17 INPUT_WIKI_URL_MESSAGE = "wiki repo(http): "
18 INPUT_BRANCH_MESSAGE = "branch: "
19 LEN_WORD_EXTENSION = 5 # .docx - 5 symbols
20 DELETE_WORD = False
21 DELETED_PICTURE = "picture"
22
23 def create_parser():
24     parser = argparse.ArgumentParser()
25     parser.add_argument('-f')
26     return parser
27
28
29 def input_cmd():
30     url = input(INPUT_URL_MESSAGE)
31     wiki_url = input(INPUT_WIKI_URL_MESSAGE)
32     branch = input(INPUT_BRANCH_MESSAGE)
33     return url, wiki_url, branch
34
35
36 def delete_directories_and_files(git, git_wiki):
37     if os.path.exists(git.local_repo):
38         shutil.rmtree(git.local_repo)
39     if os.path.exists(git_wiki.local_wiki):
40         shutil.rmtree(git_wiki.local_wiki)
41     if os.path.exists(READY_WORD) and DELETE_WORD:
42         os.remove(READY_WORD)
43     if os.path.exists(DELETED_PICTURE):
44         os.remove(DELETED_PICTURE)
45
46
47 def input_file(name):
48     with open(name) as file:
49         content = file.readlines()
50     content = [element.strip() for element in content]
51     return content[:3]
52
53
54 def main(type_of_input):
55     if type_of_input is FROM_CONSOLE:
56         url, wiki_url, branch = input_cmd()
57     else:
```

```

58     url, wiki_url, branch = input_file(type_of_input)
59     git = Gengit(url, branch)
60     git_wiki = Gengit(wiki_url)
61
62     if git.download_git() is False or git_wiki.download_git_wiki() is False:
63         delete_directories_and_files(git, git_wiki)
64         print(ERROR_MESSAGE)
65
66     word = Dword()
67     path_doc = os.path.join(git.local_repo, TIME_REPORT)
68     word.save(path_doc)
69     if word.js_content[PDF]:
70         word.convert_to_pdf(docname=path_doc)
71         git.add(PDF_EXTENSION.format(TIME_REPORT[: -LEN_WORD_EXTENSION]))
72     else:
73         git.add(TIME_REPORT)
74     git.push()
75     delete_directories_and_files(git, git_wiki)
76
77
78 if __name__ == "__main__" :
79     parser = create_parser()
80     namespace = parser.parse_args()
81
82     if not namespace.f:
83         main(FROM_CONSOLE)
84     else:
85         main(namespace.f)
86
87

```

main.py

```

1 print('main file')

```

func.py

```

1 print('func file')

```