

Санкт-Петербургский Национально Исследовательский Университет
информационных технологий, механики и оптики
Кафедра систем управления и информатики

Системное программное обеспечение

Отчет по итоговой лабораторной работе
Написание программы, используя материал курса

Работу
выполнил:
Зенкин А.М.
Группа: Р3335
Преподаватель:
Мусаев А.А.

Санкт-Петербург
2018

Содержание

1. Цель работы	2
2. Ход работы	2
2.1. Выбор среды программирования	2
2.2. Разбитие проекта на модули	3
2.3. Описание модуля pars	3
2.3.1. get_top_50	3
2.3.2. get_clans_position, get_clans_symbol_of_position, get_clans_alliance, get_clans_leader_name, get_clans_rating_symbol	5
2.3.3. get_clan_info	6
2.3.4. get_header_info	6
2.3.5. get_member_table	7
2.3.6. get_clans_statistic	8
2.3.7. get_server_stats	9
2.3.8. get_clan_stats	9
2.3.9. get_profession_stats	10
2.3.10. get_second_profession	11
2.3.11. calc_percentage	12
2.3.12. get_html_code, get_url_from_str, get_id, is_in_html_code, get_name_of_th, get_obj_info	12
2.4. Описание модуля API	14
2.4.1. clear_screen	14
2.4.2. get_full_url	14
2.4.3. get_server_number	14
2.4.4. choose_clan	14
2.4.5. print_top_50	15
2.4.6. print_clan_information	15
2.4.7. print_clan_stats	16
2.4.8. print_server_stats	16
2.4.9. print_server_pofession_stats	17
2.5. Описание модуля GoogleDocAPI	18
2.5.1. add_to_google_sheet	18
2.5.2. add_prof_server_stats	18
2.5.3. load_all_data_in_google_doc	18
2.6. Описание main	21
2.7. Примеры работы программы	23
3. Вывод	28

1. Цель работы

Написания программы для парсинга сайта с целью получения нужных данных, используя Python. После загрузки полученной информации в Google Doc.

2. Ход работы

2.1. Выбор среды программирования

В качестве среды разработки был выбран PyCharm (рис. 2.1), разработанный компанией JetBrains на основе IntelliJ IDEA. Данная среда имеет множество достоинств, некоторые из них приведены ниже:

- Статический анализ кода, подсветка синтаксиса и ошибок.
- Навигация по проекту и исходному коду: отображение файловой структуры проекта, быстрый переход между файлами, классами, методами и использованиями методов.
- Рефакторинг: переименование, извлечение метода, введение переменной, введение константы, подъем и спуск метода и т. д.
- Встроенный отладчик для Python
- Поддержка систем контроля версий: общий пользовательский интерфейс для Mercurial, Git, Subversion, Perforce и CVS с поддержкой списков изменений и слияния

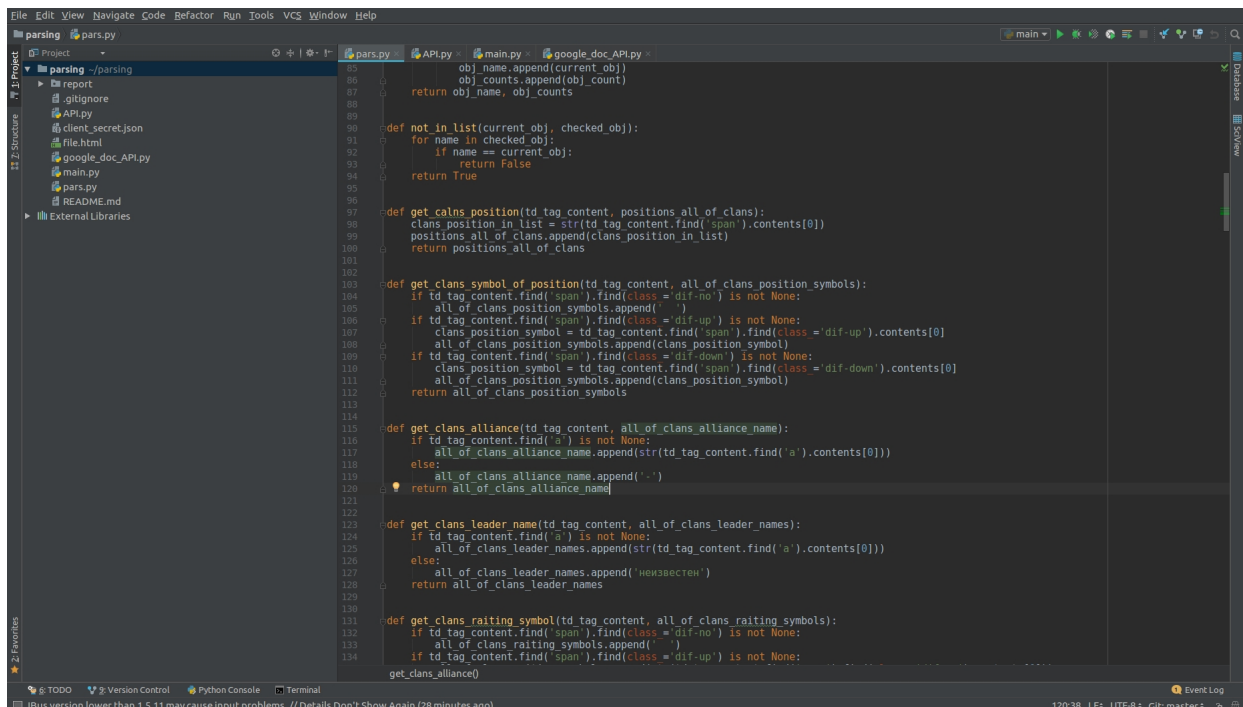


Рисунок 2.1. - интерфейс PyCharm

2.2. Разбитие проекта на модули

В ходе выполнения проекта было принято решение разбить его на следующие пакеты:

- main.py
- pars.py
- API.py
- GoogleDocAPI.py

В модуле main содержится основная программа, которая вызывает все функции из модулей.

Модуль pars содержит функции для парсинга сайта и получения информации с него.

Модуль API позволяет пользователю общаться с программой и выводить информацию в командную строку.

Модуль GoogleDocAPI осуществляет работу с Google Doc.

2.3. Описание модуля pars

2.3.1. get_top_50

Возвращает всю информацию о топ 50 кланах.

Программа 1: get_top_50

```
1 def get_top_50(top_50_clans_url):
2
3     all_of_clans_position = [ ' ' ]
4     all_of_clans_position_symbols = [ ' ' ]
5     all_of_clans_names = [ 'клан' ]
6     all_of_clans_id = []
7     all_of_clans_count = [ 'численность' ]
8     all_of_clans_alliance_names = [ 'альянс' ]
9     all_of_clans_leader_names = [ 'лидер' ]
10    all_of_clans_rating = [ 'рейтинг' ]
11    all_of_clans_raiting_symbols = [ ' ' ]
12
13    html_code = BeautifulSoup(get_html_code(top_50_clans_url), 'html.parser')
14    tbody_html = html_code.find('tbody')
15    clan_html_block = tbody_html.find_all('tr')
16
17    for blocks in clan_html_block:
18        tag_number = 1
19        td_blocks = blocks.find_all('td')
20        for td_tag_content in td_blocks:
21            if tag_number == 1:
22
23                all_of_clans_position = get_calns_position(td_tag_content,
24                                                            all_of_clans_position
25                                                            )
26                get_clans_symbol_of_position(td_tag_content,
27                                              all_of_clans_position_symbols
28                                              )
29            if tag_number == 2:
30
31                clan_name = str(blocks.find('a').contents[0])
32                all_of_clans_names.append(clan_name)
```

```

33
34         clan_id = get_id(str(blocks.find('a')))
35         all_of_clans_id.append(clan_id)
36
37         clan_count = str(get_count_from_str(str(blocks.find(class_='add'
38 ↪ ).contents[0])))
39         all_of_clans_count.append(clan_count)
40
41         if tag_number == 3:
42             all_of_clans_alliance_names = get_clans_alliance(td_tag_content,
43 ↪ all_of_clans_alliance_names
44                                     )
45         if tag_number == 4:
46             all_of_clans_leader_names = get_clans_leader_name(td_tag_content
47 ↪ ,
48 ↪ all_of_clans_leader_names
49                                     )
50         if tag_number == 5:
51             clan_rating = str(td_tag_content.find('span').contents[0])
52             all_of_clans_rating.append(clan_rating)
53
54             all_of_clans_raiting_symbols = get_clans_raiting_symbol(
55 ↪ td_tag_content,
56 ↪ all_of_clans_raiting_symbols
57                                     )
58             tag_number += 1
59
60         all_of_clans_information = [all_of_clans_position,
61 ↪ all_of_clans_position_symbols,
62                                     all_of_clans_names, all_of_clans_id,
63 ↪ all_of_clans_count,
64                                     all_of_clans_alliance_names,
65 ↪ all_of_clans_leader_names,
66                                     all_of_clans_rating,
67 ↪ all_of_clans_raiting_symbols
68                                     ]
69         return all_of_clans_information

```

2.3.2. get_clans_position, get_clans_symbol_of_position, get_clans_alliance, get_clans_leader_name, get_clans_rating_symbol

Вспомогательные функции для get_top_50. Возвращают позицию клана, маркер позиции, альянс клана, имя лидера клана и маркер рейтинга клана

Программа 2: group of functions for get_top_50

```
1 def get_clans_position(td_tag_content, positions_all_of_clans):
2     clans_position_in_list = str(td_tag_content.find('span').contents[0])
3     positions_all_of_clans.append(clans_position_in_list)
4     return positions_all_of_clans
5
6
7 def get_clans_symbol_of_position(td_tag_content, all_of_clans_position_symbols):
8     if td_tag_content.find('span').find(class_='dif-no') is not None:
9         all_of_clans_position_symbols.append(' ')
10    if td_tag_content.find('span').find(class_='dif-up') is not None:
11        clans_position_symbol = td_tag_content.find('span').find(class_='dif-up',
12    ↪ ).contents[0]
13        all_of_clans_position_symbols.append(clans_position_symbol)
14    if td_tag_content.find('span').find(class_='dif-down') is not None:
15        clans_position_symbol = td_tag_content.find('span').find(class_='dif-
16    ↪ down').contents[0]
17        all_of_clans_position_symbols.append(clans_position_symbol)
18    return all_of_clans_position_symbols
19
20 def get_clans_alliance(td_tag_content, all_of_clans_alliance_name):
21     if td_tag_content.find('a') is not None:
22         all_of_clans_alliance_name.append(str(td_tag_content.find('a').contents
23    ↪ [0]))
24     else:
25         all_of_clans_alliance_name.append('-')
26     return all_of_clans_alliance_name
27
28 def get_clans_leader_name(td_tag_content, all_of_clans_leader_names):
29     if td_tag_content.find('a') is not None:
30         all_of_clans_leader_names.append(str(td_tag_content.find('a').contents
31    ↪ [0]))
32     else:
33         all_of_clans_leader_names.append('неизвестен')
34     return all_of_clans_leader_names
35
36 def get_clans_raiting_symbol(td_tag_content, all_of_clans_raiting_symbols):
37     if td_tag_content.find('span').find(class_='dif-no') is not None:
38         all_of_clans_raiting_symbols.append(' ')
39     if td_tag_content.find('span').find(class_='dif-up') is not None:
40         all_of_clans_raiting_symbols.append(str(td_tag_content.find('span').find
41    ↪ (class_='dif-up').contents[0]))
42     if td_tag_content.find('span').find(class_='dif-down') is not None:
43         all_of_clans_raiting_symbols.append(str(td_tag_content.find('span').find
44    ↪ (class_='dif-down').contents[0]))
45     return all_of_clans_raiting_symbols
```

2.3.3. get_clan_info

Основная функция, которая возвращает всю информацию о выбранном клане.

Программа 3: get_clan_info

```
1
2 def get_clan_info(clan_id):
3     html_code = BeautifulSoup(get_html_code('http://l2on.net/?c=userdata&a=
↪ pledge&id=' + clan_id), 'html.parser')
4     clan_header_titles, clan_header_all_data = get_header_info(html_code)
5     all_members_data = get_members_table(html_code)
6
7     return clan_header_titles, clan_header_all_data, all_members_data
```

2.3.4. get_header_info

Вспомогательная функция для get_clan_info. Возвращает данные о ссылке, лидере, персонажах, профессиях, текучке, альянсе, холле, замке, крепости, "О клане" выбранного клана.

Программа 4: get_header_info

```
1 def get_header_info(html_code):
2     all_clans_header_data = []
3     header_title = []
4     tr_tag = html_code.find(class_='values').find_all('tr')
5     for number in range(len(tr_tag)):
6         td_tag_content = tr_tag[number]
7         title = str(td_tag_content.find('th').contents[0]) + ' '
8         if td_tag_content.find('td') is not None:
9             td_content = str(td_tag_content.find('td').contents[0])
10        if td_tag_content.find('a') is not None:
11            a_content = str(td_tag_content.find('a').contents[0])
12
13        if str(get_name_of_th(tr_tag[number])) == 'Ссылка:':
14            header_title.append(title)
15            all_clans_header_data.append(a_content)
16
17        if str(get_name_of_th(tr_tag[number])) == 'Лидер:':
18            header_title.append(title)
19            all_clans_header_data.append(a_content)
20
21        if str(get_name_of_th(tr_tag[number])) == 'Персонажей:':
22            header_title.append(title)
23            all_clans_header_data.append(td_content + str(td_tag_content.find(
↪ class_='add').contents[0]))
24
25        if str(get_name_of_th(tr_tag[number])) == 'Профессии:':
26            header_title.append(title)
27            all_clans_header_data.append(td_content)
28
29        if str(get_name_of_th(tr_tag[number])) == 'Текучка:':
30            header_title.append(title)
31            b_tag = td_tag_content.find_all('font')
32            all_clans_header_data.append(
33                td_content + b_tag[0].contents[0] + td_tag_content.find('td').
↪ contents[2] + b_tag[0].contents[0] +
34                td_tag_content.find('td').contents[4])
35
36        if str(get_name_of_th(tr_tag[number])) == 'Альянс:':
```

```

37         header_title.append(title)
38         all_clans_header_data.append(a_content)
39
40         if str(get_name_of_th(tr_tag[number])) == 'Холл:':
41             a_tag = td_tag_content.find_all('a')
42             header_title.append(a_tag[0].contents[0] + ': ')
43             all_clans_header_data.append(td_content + '(' + a_tag[1].contents[0]
↪ + ')')
44
45         if str(get_name_of_th(tr_tag[number])) == 'Замок:':
46             a_tag = td_tag_content.find_all('a')
47             header_title.append(str(a_tag[0].contents[0]) + ': ')
48             all_clans_header_data.append(str(a_tag[1].contents[0]) + ' (' + str(
↪ a_tag[2].contents[0]) + ')')
49
50         if str(get_name_of_th(tr_tag[number])) == 'Крепость:':
51             header_title.append(str(td_tag_content.find('td').contents[0]))
52             all_clans_header_data.append('(' + str(td_tag_content.find_all('a')
↪ [1].contents[0]) + ')')
53
54         if str(get_name_of_th(tr_tag[number])) == 'О клане:':
55             string = str(td_tag_content.select('div')[0].get_text())
56             new_string = string.split('\n')[1]
57             header_title.append(title)
58             if new_string == '':
59                 all_clans_header_data.append('Описание отсутствует')
60             else:
61                 all_clans_header_data.append(string.split('\n')[1])
62
63     return [header_title, all_clans_header_data]

```

2.3.5. get_member_table

Вспомогательная функция для get_clan_info. Возвращает всю информацию о членах выбранного клана.

Программа 5: get_member_table

```

1 def get_members_table(html_code):
2     clan_member_name = ['Имя']
3     clan_member_title = ['Титул']
4     clan_member_class = ['Класс']
5     clan_member_race = ['Раса']
6     clan_member_lvl = ['Ур.']
7     clan_member_up_date = ['Обновлен']
8     tbody_tag = html_code.find(class_='tablesorter').find('tbody')
9     tr_blocks = tbody_tag.find_all('tr')
10    for tr_block_content in tr_blocks:
11        td_blocks = tr_block_content.find_all('td')
12        tag_number = 1
13        for td_blocks_content in td_blocks:
14
15            if tag_number == 1:
16                member_name = td_blocks_content.find(class_='black').contents[0]
17                clan_member_name.append(member_name)
18                if is_title(td_blocks_content):
19                    member_title = td_blocks_content.find(class_='add').contents
↪ [0]
20            else:
21                member_title = ''

```



```

22         clan_member_title.append(member_title)
23
24     if tag_number == 2:
25         member_class = td_blocks_content.find('span').contents[0]
26         clan_member_class.append(member_class)
27
28     if tag_number == 3:
29         member_race = td_blocks_content.find('font').contents[0]
30         clan_member_race.append(member_race)
31
32     if tag_number == 4:
33         member_lv_l = td_blocks_content.contents[0]
34         clan_member_lvl.append(str(member_lv_l) + str(td_blocks_content.
↪ find('span').contents[0]))
35
36     if tag_number == 5:
37         member_up_date = td_blocks_content.find('span').contents[0]
38         clan_member_up_date.append(member_up_date)
39
40     tag_number += 1
41
42     return [clan_member_name, clan_member_title, clan_member_class,
43           clan_member_race, clan_member_lvl, clan_member_up_date]

```

2.3.6. get_clans_statistic

Возвращает статистику по клану.

Программа 6: get_clans_statistic

```

1 def get_clans_statistics(all_members_data):
2     clan_member_race = all_members_data[3]
3     clan_member_class = all_members_data[2]
4     clan_member_lvl = all_members_data[4]
5     races, race_counts = get_obj_info(clan_member_race)
6     classes, class_counts = get_obj_info(clan_member_class)
7     lvls, lvl_counts = get_obj_info(clan_member_lvl)
8     races_ratio = get_percentages(races, race_counts)
9     classes_ratio = get_percentages(classes, class_counts)
10    lvls_ratio = get_percentages(lvls, lvl_counts)
11    clan_statistics = [races, race_counts, races_ratio,
12                      classes, class_counts,
13                      classes_ratio, lvls,
14                      lvl_counts, lvls_ratio]
15    return clan_statistics

```

2.3.7. get_server_stats

Возвращает статистику по серверу.

Программа 7: get_server_stats

```
1 def get_server_stats(server_name):
2     profession_name = [], [], [], []
3     profession_count = [], [], [], []
4     site_url = 'http://l2on.net/?c=userdata&a=inf&world' + str(server_name)
5     html_code = BeautifulSoup(get_html_code(site_url), 'html.parser')
6     table_tags = html_code.find(class_='content-userdata content-userdata-inf').
↪ find_all(class_='classes')
7     table_number = 0
8     for TableTag in table_tags:
9         tr_tags = TableTag.find_all('tr')
10        for trTag in tr_tags:
11            profession = trTag.find('a').contents[0]
12            count = trTag.find(class_='count').contents[0]
13            profession_name[table_number].append(profession)
14            profession_count[table_number].append(count)
15        table_number += 1
16
17    return profession_name, profession_count
```

2.3.8. get_clan_stats

Возвращает статистику по клану.

Программа 8: get_clan_stats

```
1 def get_clans_statistics(all_members_data):
2     clan_member_race = all_members_data[3]
3     clan_member_class = all_members_data[2]
4     clan_member_lvl = all_members_data[4]
5     races, race_counts = get_obj_info(clan_member_race)
6     classes, class_counts = get_obj_info(clan_member_class)
7     lvls, lvl_counts = get_obj_info(clan_member_lvl)
8     races_ratio = get_percentages(races, race_counts)
9     classes_ratio = get_percentages(classes, class_counts)
10    lvls_ratio = get_percentages(lvls, lvl_counts)
11    clan_statistics = [races, race_counts, races_ratio,
12                      classes, class_counts,
13                      classes_ratio, lvls,
14                      lvl_counts, lvls_ratio]
15    return clan_statistics
```

2.3.9. get_profession_stats

Возвращает статистику по профессиям сервера.

Программа 9: get_profession_stats

```
1 def get_profession_stats(profession_name, profession_count):
2
3     first_profession = []
4
5     first_branch_of_the_second_profession = []
6     second_branch_of_the_second_profession = []
7     third_branch_of_the_second_profession = []
8
9     percentages_of_first_branch = []
10    percentages_of_second_branch = []
11    percentages_of_third_branch = []
12
13    first_class = ['Воитель', 'Рыцарь', 'Маг', 'Клерик', 'Светлый Рыцарь', '
    ↪ Разведчик', 'Светлый Маг', 'Оракул Евы',
14                  'Темный Рыцарь', 'Ассасин', 'Темный Маг', 'Оракул Шилен', '
    ↪ Налетчик', 'Монах', 'Шаман', 'Собираатель',
15                  'Ремесленник', 'Солдат', 'Надзиратель']
16
17    for prof in first_class:
18        professions, count, first_prof_count = get_second_profession(prof,
    ↪ profession_name, profession_count)
19        per = calc_percentage(count)
20        first_profession.append(prof)
21        prof_length = len(count)
22        first_profession.append(' ')
23        first_profession.append(' ')
24
25        first_prof_branch = ' '
26        second_prof_branch = ' '
27        third_prof_branch = ' '
28
29        first_perc_branch = ' '
30        second_perc_branch = ' '
31        third_perc_branch = ' '
32
33        if prof_length == 3:
34            first_prof_branch = professions[0]
35            first_perc_branch = per[0]
36            second_prof_branch = professions[1]
37            second_perc_branch = per[1]
38            third_prof_branch = professions[2]
39            third_perc_branch = per[2]
40
41        if prof_length == 2:
42            first_prof_branch = professions[0]
43            first_perc_branch = per[0]
44            second_prof_branch = professions[1]
45            second_perc_branch = per[1]
46
47        if prof_length == 1:
48            first_prof_branch = professions[0]
49            first_perc_branch = per[0]
50
51        first_branch_of_the_second_profession.append(first_prof_branch)
52        second_branch_of_the_second_profession.append(second_prof_branch)
53        third_branch_of_the_second_profession.append(third_prof_branch)
```

```

54     percentages_of_first_branch.append(first_perc_branch)
55     percentages_of_second_branch.append(second_perc_branch)
56     percentages_of_third_branch.append(third_perc_branch)
57
58
59     second_profession = [first_branch_of_the_second_profession,
60                          second_branch_of_the_second_profession,
61                          third_branch_of_the_second_profession]
62
63     percentages = [percentages_of_first_branch,
64                   percentages_of_second_branch,
65                   percentages_of_third_branch]
66
67     return first_profession, second_profession, percentages

```

2.3.10. get_second_profession

Вспомогательная функция для GetProfessionStats.

Программа 10: get_second_profession

```

1 def get_second_profession(first_prof, all_profession_names,
2   ↪ all_profession_counts):
3     classes = { 'Воитель': ( 'Копейщик', 'Гладиатор' ),
4                  'Рыцарь': ( 'Паладин', 'Мститель' ),
5                  'Маг': ( 'Властитель Огня', 'Некромант', 'Колдун' ),
6                  'Клерик': ( 'Епископ', 'Проповедник' ),
7                  'Светлый Рыцарь': ( 'Рыцарь Евы', 'Менестрель' ),
8                  'Разведчик': ( 'Следопыт', 'Серебряный Рейнджер' ),
9                  'Светлый Маг': ( 'Певец Заклинаний', 'Последователь Стихий' ),
10                 'Оракул Евы': [ 'Мудрец Евы' ],
11                 'Темный Рыцарь': ( 'Рыцарь Шилен', 'Танцор Смерти' ),
12                 'Ассасин': ( 'Странник Бездны', 'Призрачный Рейнджер' ),
13                 'Темный Маг': ( 'Заклинатель Ветра', 'Последователь Тьмы' ),
14                 'Оракул Шилен': [ 'Мудрец Шилен' ],
15                 'Налетчик': [ 'Разрушитель' ],
16                 'Монах': [ 'Отшельник' ],
17                 'Шаман': ( 'Верховный Шаман', 'Вестник Войны' ),
18                 'Собиратель': [ 'Охотник за Наградой' ],
19                 'Ремесленник': [ 'Кузнец' ],
20                 'Солдат': ( 'Берсерк', 'Палач' ),
21                 'Надзиратель': ( 'Палач', 'Арбалетчик' ),
22                 'Воин Артеи': [ 'Боец Сайхи' ],
23                 'Маг Артеи': [ 'Последователь Сайхи' ] }
24     first_prof_count = all_profession_counts[3][all_profession_names[3].index(
25   ↪ first_prof)]
26     professions = []
27     count = []
28     second_prof_list = all_profession_names[2]
29     second_prof = classes[first_prof]
30     for prof in range(len(second_prof)):
31         position_in_list = second_prof_list.index(second_prof[prof])
32         professions.append(second_prof[prof])
33         count.append(all_profession_counts[2][position_in_list])
34     return professions, count, first_prof_count

```

2.3.11. calc_percentage

Возвращает проценты по количеству профессий.

Программа 11: calc_percentage

```
1 def calc_percentage(count):
2     total = int(0)
3     percentage = []
4     for number in range(len(count)):
5         total += int(count[number])
6     for number in range(len(count)):
7         calc = int(count[number]) / total
8         percentage.append(round(calc * 100, 2))
9     return percentage
```

2.3.12. get_html_code, get_url_from_str, get_id, is_in_html_code, get_name_of_th, get_obj_info

Вспомогательные функции.

Программа 12: get_html_code

```
1
2 def get_html_code(site_url):
3     open_url = urllib.request.urlopen(site_url)
4     html_code = open_url.read()
5     return html_code
```

Программа 13: get_url_from_str

```
1
2 def get_url_from_str(string):
3     return string[string.find('/') : string.rfind('")] ]
```

Программа 14: get_id

```
1
2 def get_id(string):
3     return string[string.find('id=') + 3 : string.rfind('")] ]
```

Программа 15: is_in_html_code

```
1
2 def is_in_html_code(html_code, obj):
3     flag = False
4     for content in html_code.find(class_='values').find_all('a'):
5         if content.contents[0] is not None:
6             content = content.contents[0]
7             if str(content) == str(obj):
8                 flag = True
9     return flag
```

Программа 16: get_name_of_th

```

1
2 def get_name_of_th(html_code):
3     th_code = html_code.find('th')
4     if th_code.find('a') is not None:
5         return str(th_code.find('a').contents[0] + ':')
6     else:
7         return str(th_code.contents[0])

```

Программа 17: get_obj_info

```

1
2 def get_obj_info(clan_member_obj):
3     checked_obj = []
4     obj_name = []
5     obj_counts = []
6     for number_i in range(len(clan_member_obj) - 1):
7         obj_count = 0
8         current_obj = clan_member_obj[number_i + 1]
9         if not_in_list(current_obj, checked_obj):
10            checked_obj.append(current_obj)
11            for number_j in range(len(clan_member_obj)):
12                if clan_member_obj[number_j] == current_obj:
13                    obj_count += 1
14            obj_name.append(current_obj)
15            obj_counts.append(obj_count)
16    return obj_name, obj_counts

```

Программа 18: is_title

```

1
2 def is_title(td_blocks_content):
3     if td_blocks_content.find(class_='add') is not None:
4         return True
5     else:
6         return False

```

Программа 19: not_in_list

```

1
2 def not_in_list(current_obj, checked_obj):
3     for name in checked_obj:
4         if name == current_obj:
5             return False
6     return True

```

2.4. Описание модуля API

2.4.1. clear_screen

Очищает экран командной строки.

Программа 20: clear_screen

```
1
2 def clear_screen():
3     if sys.platform == 'win32':
4         os.system('cls')
5     else:
6         os.system('clear')
```

2.4.2. get_full_url

Возвращает полный путь к странице.

Программа 21: get_full_url

```
1
2 def get_full_url(server_number):
3     return str(URL + ServersName[int(server_number) - 1])
```

2.4.3. get_server_number

Позволяет пользователю выбрать желаемый сервер.

Программа 22: get_server_number

```
1
2 def get_server_number():
3     clear_screen()
4
5     print('##### HELLO MY FRIEND #####'.center(100))
6     print("##### LET's START PARSING #####".center(100))
7     print('Choose server: '.center(100))
8     numb = 1
9     for ServerName in ServersName:
10        print(('(' + str(numb) + ') ' + ServerName).center(100))
11        numb += 1
12
13    server_number = input('Server number _'.rjust(30))
14
15    return int(server_number)
```

2.4.4. choose_clan

Позволяет пользователю выбрать желаемый клан.

Программа 23: choose_clan

```
1
2 def choose_clan():
3     print('')
4     selected_clan_number = input('Choose clan( 1-50):'.rjust(30))
5     if IsOutputToConsole:
6         clear_screen()
7
8     return int(selected_clan_number)
```

2.4.5. print_top_50

Выводит в командную строку топ 50 кланов.

Программа 24: print_top_50

```
1
2 def print_top_50(top_50_clans_information, server_number):
3     if IsOutputToConsole:
4         print(('##### Top 50 clans of ' + ServersName[int(server_number)
5         ↪ - 1] + ' #####').center(100))
6         for i in range(51):
7             print(top_50_clans_information[0][i].center(4),
8             ↪ top_50_clans_information[1][i].rjust(4),
9             ↪ top_50_clans_information[2][i].ljust(15),
10            ↪ top_50_clans_information[4][i].center(15),
11            ↪ top_50_clans_information[5][i].ljust(20),
12            ↪ top_50_clans_information[6][i].ljust(20),
13            ↪ top_50_clans_information[7][i].ljust(5),
14            ↪ top_50_clans_information[8][i])
```

2.4.6. print_clan_information

Выводит в командную строку информацию о выбранном клане.

Программа 25: print_clan_information

```
1
2 def print_clan_information(clan_header_title, clan_header_data,
3                             table_with_member_data, top_50_clans_information,
4                             selected_clan_number):
5     if IsOutputToConsole:
6
7         member_names = table_with_member_data[0]
8         member_titles = table_with_member_data[1]
9         member_classes = table_with_member_data[2]
10        member_races = table_with_member_data[3]
11        member_lvls = table_with_member_data[4]
12        member_updates = table_with_member_data[5]
13
14        name_of_the_selected_clan = top_50_clans_information[2]
15        print(('##### '
16        ↪ + name_of_the_selected_clan[int(selected_clan_number)]
17        ↪ + "'s CLAN INFO #####"
18        ↪ ).center(100)
19        ↪ )
20        for i in range(len(clan_header_title)):
21            print('          ' + clan_header_title[i] + ', ' +
22            ↪ clan_header_data[i])
23            print('##### COCTAB #####'.center(100))
24            for i in range(len(table_with_member_data[0])):
25                print(''.ljust(4) + member_names[i].ljust(20) + member_titles[i].
26                ↪ ljust(20) + member_classes[i].ljust(30) +
27                ↪ member_races[i].ljust(15) + member_lvls[i].ljust(10) +
28                ↪ member_updates[i].ljust(20))
```


2.4.7. print_clan_stats

Выводит в командную строку статистику клана.

Программа 26: print_clan_stats

```
1
2 def print_clan_stats(clan_races, clan_race_counts, clan_races_ratio,
3                     clan_classes, clan_class_counts, clan_classes_ratio,
4                     clan_lvls, clan_lvl_counts, clan_lvls_ratio):
5     if IsOutputToConsole:
6         print('')
7         print('##### Races Stats #####'.center(100))
8         for i in range(len(clan_races)):
9             print(str(clan_races[i]).ljust(30) + str(clan_race_counts[i]).ljust
10 ↪ (15) + str(clan_races_ratio[i]))
11         print('')
12         print('##### Classes Stats #####'.center(100))
13         for i in range(len(clan_classes)):
14             print(str(clan_classes[i]).ljust(30) + str(clan_class_counts[i]).
15 ↪ ljust(15) + str(clan_classes_ratio[i]))
16         print('')
17         print('##### LvLs Stats #####'.center(100))
18         for i in range(len(clan_lvls)):
19             print(str(clan_lvls[i]).ljust(30) + str(clan_lvl_counts[i]).ljust
20 ↪ (15) + str(clan_lvls_ratio[i]))
```

2.4.8. print_server_stats

Выводит в командную строку статистику по серверу.

Программа 27: print_server_stats

```
1
2 def print_server_stats(all_profession_names, profession_count):
3     if IsOutputToConsole:
4         professions = ['Четвёртые', 'Третьи', 'Вторые', 'Первые']
5         for i in range(len(all_profession_names)):
6             print('')
7             print(('##### ' + professions[i] + ' профессии' +
8 ↪ #####').center(100))
9             print('')
10            for j in range(len(all_profession_names[i])):
11                print('' + (str(all_profession_names[i][j]).ljust(30) + ' ' +
12 ↪ str(profession_count[i][j]).rjust(
13 ↪ 5)).center(100))
```

2.4.9. print_server_pofession_stats

Выводит в командную строку статистику по профессиям сервера.

Программа 28: print_server_pofession_stats

```
1
2 def print_server_pofession_stats(first_profession , second_profession ,
  ↪ percentages):
3     if IsOutputToConsole:
4         for i in range(len(second_profession[0])):
5             print(first_profession[i*3])
6             print(second_profession[0][i].ljust(25) + ' ' ,
7                   + second_profession[1][i].ljust(25) + ' ' ,
8                   + second_profession[2][i].ljust(25)
9                   )
10            print(str(percentages[0][i]).ljust(25) + ' ' ,
11                  + str(percentages[1][i]).ljust(25) + ' ' ,
12                  + str(percentages[2][i]).ljust(25)
13                  )
14            print('')
```

2.5. Описание модуля GoogleDocAPI

2.5.1. add_to_google_sheet

Добавляет данный в Google Doc.

Программа 29: add_to_google_sheet

```
1
2 def add_to_google_sheet(sheet_name, worksheet_name, data_list, start_column,
  ↪ row_offset=0, column_offset=0):
3     list_length = len(data_list)
4     sheet_data = client.open(sheet_name).worksheet(worksheet_name)
5     for row in range(list_length):
6         sheet_data.update_cell(row + 1 + row_offset, start_column +
  ↪ column_offset, data_list[row])
```

2.5.2. add_prof_server_stats

Добавляет статистику по профессиям сервера в Google Doc.

Программа 30: add_prof_server_stats

```
1
2 def add_prof_server_stats(first_profession, second_profession, percentages):
3     sheet_name = 'l2onParsing'
4     worksheet_name = 'ProfServerStats'
5     sheet_data = client.open(sheet_name).worksheet(worksheet_name)
6     list_length = len(first_profession)
7     for row in range(list_length):
8         sheet_data.update_cell(row + 1, 1, first_profession[row])
9     row = 1
10    for i in range(len(second_profession[0])):
11        column = 2
12        for j in range(3):
13            sheet_data.update_cell(row, column, second_profession[j][i])
14            column += 1
15        row += 3
16    row = 2
17    for i in range(len(percentages[0])):
18        column = 2
19        for j in range(3):
20            sheet_data.update_cell(row, column, percentages[j][i])
21            column += 1
22        row += 3
```

2.5.3. load_all_data_in_google_doc

Загружает все в Google Doc.

Программа 31: load_all_data_in_google_doc

```
1
2 def load_all_data_in_google_doc(clan_header, all_member_data,
3     races_data, classes_data,
4     clan_lvl_data, first_profession_data,
5     second_profession_data, third_profession_data,
6     fourth_profession_data, first_profession,
7     second_profession, percentages):
8     for list_count in range(len(clan_header)):
```

```

9         sheet_name = 'l2onParsing'
10        worksheet_name = 'ClanStats'
11        add_to_google_sheet(sheet_name, worksheet_name, clan_header[list_count],
↪ list_count + 1)
12
13    for list_count in range(len(all_member_data)):
14        row_offset = 12
15        sheet_name = 'l2onParsing'
16        worksheet_name = 'ClanStats'
17        add_to_google_sheet(sheet_name, worksheet_name, all_member_data[
↪ list_count],
18                               list_count + 1, row_offset)
19
20    for list_count in range(len(races_data)):
21        sheet_name = 'l2onParsing'
22        worksheet_name = 'ClanMemberStats'
23        start_row = list_count + 1
24        add_to_google_sheet(sheet_name, worksheet_name, races_data[list_count],
↪ start_row)
25
26    for list_count in range(len(classes_data)):
27        sheet_name = 'l2onParsing'
28        worksheet_name = 'ClanMemberStats'
29        start_row = list_count + 1
30        row_offset = 0
31        column_offset = 4
32        add_to_google_sheet(sheet_name, worksheet_name, classes_data[list_count
↪ ],
33                               start_row, row_offset, column_offset)
34
35    for list_count in range(len(clan_lvl_data)):
36        sheet_name = 'l2onParsing'
37        worksheet_name = 'ClanMemberStats'
38        start_row = list_count + 1
39        row_offset = 0
40        column_offset = 8
41        add_to_google_sheet(sheet_name, worksheet_name, clan_lvl_data[list_count
↪ ],
42                               start_row, row_offset, column_offset)
43
44    for list_count in range(len(first_profession_data)):
45        sheet_name = 'l2onParsing'
46        worksheet_name = 'ServerStats'
47        start_row = list_count + 1
48        row_offset = 1
49        column_offset = 0
50        add_to_google_sheet(sheet_name, worksheet_name, first_profession_data[
↪ list_count],
51                               start_row, row_offset, column_offset)
52
53    for list_count in range(len(second_profession_data)):
54        sheet_name = 'l2onParsing'
55        worksheet_name = 'ServerStats'
56        start_row = list_count + 1
57        row_offset = 1
58        column_offset = 3
59        add_to_google_sheet(sheet_name, worksheet_name, second_profession_data[
↪ list_count],
60                               start_row, row_offset, column_offset)
61

```

```

62     for list_count in range(len(third_profession_data)):
63         sheet_name = 'l2onParsing'
64         worksheet_name = 'ServerStats'
65         start_row = list_count + 1
66         row_offset = 1
67         column_offset = 6
68         add_to_google_sheet(sheet_name, worksheet_name, third_profession_data[
↪ list_count],
69                             start_row, row_offset, column_offset)
70
71     for list_count in range(len(fourth_profession_data)):
72         sheet_name = 'l2onParsing'
73         worksheet_name = 'ServerStats'
74         start_row = list_count + 1
75         row_offset = 1
76         column_offset = 9
77         add_to_google_sheet(sheet_name, worksheet_name, fourth_profession_data[
↪ list_count],
78                             start_row, row_offset, column_offset)
79
80     add_prof_server_stats(first_profession, second_profession, percentages)

```

2.6. Описание main

Программа 32: main

```
1
2 import pars as parsing
3 import API
4 import GoogleDocAPI as GoogleDoc
5
6 server_number = API.get_server_number()
7 full_url_name = API.get_full_url(server_number)
8
9 top_50_clans_information = parsing.get_top_50(full_url_name)
10 API.print_top_50(top_50_clans_information, server_number)
11
12 id_of_clans = top_50_clans_information[3]
13 selected_clan_number = API.choose_clan()
14 clan_id = id_of_clans[selected_clan_number-1]
15
16 clan_header_title, clan_header_data, table_with_member_data = parsing.
    ↪ get_clan_info(clan_id)
17
18 clan_header = [clan_header_title, clan_header_data]
19
20 API.print_clan_information(clan_header_title,
21                             clan_header_data,
22                             table_with_member_data,
23                             top_50_clans_information,
24                             selected_clan_number
25                             )
26
27 member_names = table_with_member_data[0]
28 member_titles = table_with_member_data[1]
29 member_classes = table_with_member_data[2]
30 member_races = table_with_member_data[3]
31 member_lvls = table_with_member_data[4]
32 member_updates = table_with_member_data[5]
33
34 all_member_data = [member_names, member_titles, member_classes, member_races,
    ↪ member_lvls, member_updates]
35
36 ClanStatistics = parsing.get_clans_statistics(all_member_data)
37
38 clan_races = ['Рассы']
39 clan_race_counts = ['Количество']
40 clan_races_ratio = ['Процент']
41 clan_races.extend(ClanStatistics[0])
42 clan_race_counts.extend(ClanStatistics[1])
43 clan_races_ratio.extend(ClanStatistics[2])
44
45 races_data = [clan_races, clan_race_counts, clan_races_ratio]
46
47 clan_classes = ['Классы']
48 clan_class_counts = ['Количество']
49 clan_classes_ratio = ['Процент']
50 clan_classes.extend(ClanStatistics[3])
51 clan_class_counts.extend(ClanStatistics[4])
52 clan_classes_ratio.extend(ClanStatistics[5])
53
54 classes_data = [clan_classes, clan_class_counts, clan_classes_ratio]
55
```

```

56 clan_lvls = [ 'Лвл' ]
57 clan_lvl_counts = [ 'Количество' ]
58 clan_lvls_ratio = [ 'Проценты' ]
59 clan_lvls.extend(ClanStatistics[6])
60 clan_lvl_counts.extend(ClanStatistics[7])
61 clan_lvls_ratio.extend(ClanStatistics[8])
62
63 clan_lvl_data = [clan_lvls, clan_lvl_counts, clan_lvls_ratio]
64
65 API.print_clan_stats(clan_races, clan_race_counts, clan_races_ratio,
66                     clan_classes, clan_class_counts, clan_classes_ratio,
67                     clan_lvls, clan_lvl_counts, clan_lvls_ratio)
68
69 all_profession_names, profession_count = parsing.get_server_stats(server_number
    ↪ -1)
70
71 first_profession_name = all_profession_names[3]
72 second_profession_name = all_profession_names[2]
73 third_profession_name = all_profession_names[1]
74 fourth_profession_name = all_profession_names[0]
75
76 first_profession_counts = profession_count[3]
77 second_profession_counts = profession_count[2]
78 third_profession_counts = profession_count[1]
79 fourth_profession_counts = profession_count[0]
80
81 first_profession_data = [first_profession_name, first_profession_counts]
82 second_profession_data = [second_profession_name, second_profession_counts]
83 third_profession_data = [third_profession_name, third_profession_counts]
84 fourth_profession_data = [fourth_profession_name, fourth_profession_counts]
85
86 API.print_server_stats(all_profession_names, profession_count)
87
88 first_profession, second_profession, percentages = parsing.get_profession_stats(
    ↪ all_profession_names,
89
    ↪ profession_count)
90
91 API.print_server_profession_stats(first_profession, second_profession,
    ↪ percentages)
92
93 GoogleDoc.load_all_data_in_google_doc(clan_header, all_member_data,
94                                       races_data, classes_data,
95                                       clan_lvl_data, first_profession_data,
96                                       second_profession_data,
    ↪ third_profession_data,
97                                       fourth_profession_data, first_profession,
98                                       second_profession, percentages)

```

2.7. Примеры работы программы

Примеры работы программы приведены на рисунках 2.2 - 2.11.

```
##### HELLO MY FRIEND #####
##### LET's START PARSING #####
Choose server:
(1)Blackbird
(2)Esthus
(3)Airin
(4)Elcardia

Server number 4
```

Рисунок 2.2. - выбор сервера

##### Top 50 clans of Elcardia #####					
	клан	численность	альянс	лидер	рейтинг
1.	RiseOfAbyss	619	RiseofAbyss	неизвестен	14841 ▼
2.	Freedoms	129	Eclipse	MONctrik	4285 ▲
3.	NoFear	122	Eclipse	неизвестен	3633 ▼
4.	URAL	121	ЕКБ	Cripto	3521 ▲
5.	▲ Furiuos	94	Federation	неизвестен	2763 ▼
6.	▲ BlackNight	99	Ola	МилаяПоночка	2750 ▼
7.	▲ BadCompany	115	BAD	КритыРешают	2605 ▲
8.	▲ Молодость	92	TradeFederation	Ватрушечка	2598 ▲
9.	▲ Vizantia	94	Благие	Noumak	2568 ▼
10.	▼ Squad	87	Federation	неизвестен	2551 ▼
11.	▲ Норманны	107	Normandy	неизвестен	2549 ▲
12.	▲ Siberia	89	Accord	Barneo	2541 ▲
13.	▲ WhiteTemple	111	Whiteguadians	неизвестен	2498 ▲
14.	▼ НоуПроблемс	87	ДРАЗЗИЛЫ	неизвестен	2470 ▼
15.	▲ SoldierOfFortune43	108	-	неизвестен	2432
16.	▲ BlackoutEmpire	91	RiseofAbyss	неизвестен	2432 ▲
17.	▲ Archons	84	Greatness	неизвестен	2392 ▼
18.	▲ Gladiolus	78	ЭПИТЕТ	ДобраяНяшка	2344 ▲
19.	▲ Готей13	77	Piraniy	КРАСОТКА77	2343 ▼
20.	▲ Bse	85	Deepers	неизвестен	2334 ▲
21.	▼ ImperialLegion	83	Independence	Pedor	2322 ▼
22.	▲ Chaotics	98	Deepers	неизвестен	2309 ▲
23.	▼ КрестовыйПоход	82	SODALITY	неизвестен	2293 ▼
24.	▼ Swords	78	Superbia	неизвестен	2293 ▼
25.	▼ SpecialForces	100	PVE	неизвестен	2238 ▼
26.	▼ DELOR	74	Juniors	ShadowSouls	2188 ▼
27.	▼ BAPГ	79	IceDozen	неизвестен	2112 ▼
28.	▼ RaidBoss	92	East	неизвестен	2108 ▼
29.	▼ BattleBorn	85	Metallica	неизвестен	2100 ▼
30.	▲ ГильдияХвостФей	76	-	неизвестен	2023 ▲
31.	▲ Era	74	Hope	iPadre	2013 ▲
32.	▲ Эдем	71	Олимп	неизвестен	1984 ▲
33.	▲ Gotika72	77	SSS	неизвестен	1972 ▲
34.	▲ Astflon	82	Астрос	неизвестен	1960 ▲
35.	▼ Metallica	76	Metallica	неизвестен	1936 ▼
36.	▲ Hunter	79	-	ГномКузька	1888 ▼
37.	▼ 42Регион	73	SpiritOfLife	неизвестен	1864 ▼
38.	▲ Амбrella43	76	Underwater	неизвестен	1855 ▲
39.	▲ Midnight	65	Atlas	неизвестен	1854 ▲
40.	▼ spartanci	75	Sparta	неизвестен	1839 ▼

Рисунок 2.3. - топ 50 кланов

Choose clan(1-50):3

NoFear's CLAN INFO

Ссылка: адрес не указан
Персонажей: 122 (115 дворян и 1 герой)
Профессии: 1 персонаж с третьей (1%) и 121 с четвертой (99%)
Текучка: за последнюю неделю 4 персонажа вступило в клан и 4 покинуло
Альянс: Eclipse
0 клане: Описание отсутствует

СОСТАВ

Имя	Титул	Класс	Раса	Ур.	Обновлен
AllHail		Диверсант Эура	Камаэль	99+	день назад
AchlleSS		Глас Судьбы Иса	Человек	99+	2 дня назад
Adarin	Freelancer	Страж Теней Эура	Тёмный эльф	99+	день назад
anakimka		Владыка Теней Веньо	Тёмный эльф	103+	день назад
ArsMoriendi	медоед	Заступник Сайхи	Артея	101+	день назад
BlackSpot	РАБ ДИМОНА	Архимаг Фео	Тёмный эльф	104+	3 дня назад
Buxants0v	ЗАНЯТО 24\7	Страж Лунного Света Эура	Светлый эльф	99+	4 дня назад
Caduceus	Champi	Заступник Сайхи	Артея	99+	день назад
Clem		Мастер Тира	Гном	99+	4 дня назад
coder1		Храмовник Шилен	Тёмный эльф	99+	неделю назад
CrazYMisS		Повелитель Бури Фео	Тёмный эльф	99+	день назад
CyceJl		Странник Ветра Одала	Человек	102+	час назад
DarkJIily		Владыка Теней Веньо	Тёмный эльф	99+	день назад
DeusD	Sunny TM	Титан Тира	Орк	99+	2 дня назад
DominikToretto		Призрачный Охотник Одала	Тёмный эльф	102+	2 часа назад
ElvenRx		Странник Ветра Одала	Светлый эльф	102+	день назад
EpicDramma		Страж Лунного Света Эура	Светлый эльф	99+	день назад
EveliYa		Виртуоз Иса	Светлый эльф	99+	2 дня назад
Fisti	Satisfaction	Жрец Евы Альгиза	Светлый эльф	99+	день назад
Grinderella		Заступник Сайхи	Артея	102+	2 дня назад
hallllllllllk		Титан Тира	Орк	99+	неделю назад
HellBell		Деспот Иса	Орк	102+	день назад
Hicari		Диверсант Эура	Камаэль	99+	день назад
iFartMan		Призрачный Танцор Иса	Тёмный эльф	99+	день назад
IIqOpII		Заступник Сайхи	Артея	99+	4 дня назад
iMad		Апостол Иса	Человек	101+	день назад
iMONctrik		Рыцарь Феникса Сигеля	Светлый эльф	99+	день назад
iMureno		Страж Теней Эура	Орк	99+	2 часа назад
Inserta		Жрец Евы Альгиза	Светлый эльф	99+	4 дня назад
Inwers	Riskbreaker	Дуэлист Тира	Человек	99+	день назад
IsRapsody		Глас Судьбы Иса	Орк	99+	день назад
iSunny		Страж Теней Эура	Тёмный эльф	100+	день назад
Jast1n		Пожиратель Душ Фео	Светлый эльф	100+	неделю назад
jXc		Храмовник Шилен Сигеля	Тёмный эльф	99+	неделю назад
KATOK	-Moped78-	Рыцарь Ада Сигеля	Человек	100+	3 дня назад
kop175	Гриндевальт	Снайпер Эура	Человек	100+	день назад
KuroiGetsuga		Владыка Теней Веньо	Человек	99+	2 дня назад
LastNoldor		Храмовник Шилен Сигеля	Тёмный эльф	99+	день назад
Ligeya		Мастер Стихий Веньо	Светлый эльф	99+	23 часа назад

Рисунок 2.4. - информация о выбранном клане

Races Stats

Рассы	Количество	Процент
Камаэль	11	9.02%
Человек	21	17.21%
Тёмный эльф	32	26.23%
Артея	16	13.11%
Светлый эльф	22	18.03%
Гном	5	4.1%
Орк	15	12.3%

Рисунок 2.5. - статистика по рассам

Классы	Количество	Процент
Диверсант Эура	3	2.46%
Глас Судьбы Иса	4	3.28%
Страж Теней Эура	6	4.92%
Владыка Теней Веньо	7	5.74%
Заступник Сайхи	15	12.3%
Архимаг Фео	1	0.82%
Страж Лунного Света Эура	4	3.28%
Мастер Тира	3	2.46%
Храмовник Шилен	1	0.82%
Повелитель Бури Фео	4	3.28%
Странник Ветра Одала	3	2.46%
Титан Тира	11	9.02%
Призрачный Охотник Одала	3	2.46%
Виртуоз Иса	4	3.28%
Жрец Евы Альгиза	7	5.74%
Деспот Иса	1	0.82%
Призрачный Танцор Иса	5	4.1%
Апостол Иса	4	3.28%
Рыцарь Феникса Сигеля	2	1.64%
Дуэлист Тира	3	2.46%
Пожиратель Душ Фео	3	2.46%
Храмовник Шилен Сигеля	5	4.1%
Рыцарь Ада Сигеля	3	2.46%
Снайпер Эура	2	1.64%
Мастер Стихий Веньо	1	0.82%
Кардинал Альгиза	5	4.1%
Жрец Шилен Альгиза	4	3.28%
Кладоискатель Одала	1	0.82%
Каратель Тира	2	1.64%
Авантюрист Одала	1	0.82%
Инквизитор Фео	3	2.46%
Магистр Магии Фео	1	0.82%

Рисунок 2.6. - статистика по классам

Лвл	Количество	Проценты
99+	84	68.85%
103+	1	0.82%
101+	8	6.56%
104+	1	0.82%
102+	13	10.66%
100+	11	9.02%
85+	4	3.28%

Рисунок 2.7. - статистика по лвлу

##### Четвёртые профессии #####		
Владыка Теней Веньо		11455
Заступник Сайхи		10525
Титан Тира		8968
Призрачный Танцор Иса		8133
Громовержец Сайхи		8114
Кладоискатель Одала		7795
Мастер Тира		7640
Апостол Иса		6806
Глас Судьбы Иса		6726
Призрачный Охотник Одала		5992
Повелитель Бури Фео		5623
Страж Теней Эура		5608
Мастер Стихий Веньо		4901
Жрец Шилен Альгиза		4877
Каратель Тира		4864
Жрец Евы Альгиза		4773
Кардинал Альгиза		4722
Чернокнижник Веньо		4269
Снайпер Эура		3703
Виртуоз Иса		3503
Полководец Тира		3293
Храмовник Шилен Сигеля		3218
Авантюрист Одала		3190
Диверсант Эура		3098
Пожиратель Душ Фео		3062
Дуэлист Тира		3023
Инквизитор Фео		2926
Деспот Иса		2789
Страж Лунного Света Эура		2536
Заклинатель Иса		2247
Аватар Тира		2200
Странник Ветра Одала		2145

Рисунок 2.8. - статистика по 4 профессиям

Воитель	Гладиатор	
Копейщик		
30.96	69.04	
Рыцарь	Мститель	
Паладин		
54.77	45.23	
Маг	Некромант	Колдун
Властитель Огня		
46.23	30.19	23.58
Клерик	Проповедник	
Епископ		
51.26	48.74	
Светлый Рыцарь	Менестрель	
Рыцарь Евы		
37.71	62.29	
Разведчик	Серебряный Рейнджер	
Следопыт		
37.13	62.87	
Светлый Маг	Последователь Стихий	
Певец Заклинаний		
61.11	38.89	
Оракул Евы		
Мудрец Евы		
100.0		
Темный Рыцарь	Танцор Смерти	
Рыцарь Шилен		
36.41	63.59	

Рисунок 2.9. - статистика по выбору профессий

	A	B	C	D	E	F
1	Ссылка:	адрес не указан				
2	Персонажей:	122 (115 дворян и 1 герой)				
3	Профессии:	1 персонаж с третьей (1%) и 121 с четвертой (99%)				
4	Текучка:	за последнюю неделю 4 персонажа вступило в клан и 4 покинуло				
5	Альянс:	Eclipse				
6	О клане:	Описание отсутствует				
7						
8						
9						
10						
11						
12						
13	Имя	Титул	Класс	Раса	Ур.	Обновлен
14	A11Hail		Диверсант Эура	Камаэль	99+	день назад
15	AchlleSS		Глас Судьбы Иса	Человек	99+	2 дня назад
16	Adarin	Freelancer	Страж Теней Эура	Тёмный эльф	99+	день назад
17	anakimka		Владыка Теней Веньо	Тёмный эльф	103+	день назад
18	ArsMoriendi	медоед	Заступник Сайхи	Артея	101+	день назад
19	BlackSpot	РАБ ДИМОНА	Архимаг Фео	Тёмный эльф	104+	3 дня назад
20	Buxants0v	ЗАНЯТО 24\7	Страж Лунного Света Эура	Светлый эльф	99+	4 дня назад
21	Caduceus	Champi	Заступник Сайхи	Артея	99+	день назад
22	Clem		Мастер Тира	Гном	99+	4 дня назад
23	coder1		Храмовник Шилен	Тёмный эльф	99+	неделю назад
24	CrazyMisS		Повелитель Бури Фео	Тёмный эльф	99+	день назад
25	CyseJI		Странник Ветра Одала	Человек	102+	час назад
26	DarKJlily		Владыка Теней Веньо	Тёмный эльф	99+	день назад
27	DeusD	Sunny TM	Титан Тира	Орк	99+	2 дня назад
28	DominikToretto		Призрачный Охотник Одала	Тёмный эльф	102+	2 часа назад
29	ElvenRx		Странник Ветра Одала	Светлый эльф	102+	день назад
30	EpicDramma		Страж Лунного Света Эура	Светлый эльф	99+	день назад
31	Eveliya		Виртуоз Иса	Светлый эльф	99+	2 дня назад
32	Fisti	Satisfaction	Жрец Евы Альгиза	Светлый эльф	99+	день назад
33	Grinderella		Заступник Сайхи	Артея	102+	2 дня назад

Рисунок 2.10. - пример вывода данных в Google Doc

	A	B	C	D	E	F	G	H	I	J	K
1	Первая профессия			Вторая профессия			Третья профессия			Четвертая профессия	
2	Темный Маг	7229		Охотник за Наградой	22154		Храмовник Шилен	3280		Владыка Теней Веньо	11455
3	Собиратель	4490		Кузнец	13457		Укротитель Ветра	2131		Заступник Сайхи	10522
4	Ассасин	3536		Боец Сайхи	13271		Кладомщикатель	1752		Титан Тира	8965
5	Воитель	3267		Берсерк	12131		Глас Судьбы	1744		Призрачный Танцор Иса	8132
6	Темный Рыцарь	2976		Последователь Сайхи	11844		Титан	1550		Громовержец Сайхи	8114
7	Солдат	2899		Заклинатель Ветра	9482		Владыка Теней	1549		Кладомщикатель Одала	7794
8	Шаман	2840		Гладиатор	8821		Ловец Ветра	1397		Мастер Тира	7637
9	Маг	2761		Танцор Смерти	8793		Повелитель Бури	1351		Апостол Иса	6805
10	Ремесленник	2677		Разрушитель	8083		Мастер	1224		Глас Судьбы Иса	6726
11	Налетчик	2537		Последователь Тьмы	7724		Каратель	1187		Призрачный Охотник Одала	5991
12	Надзиратель	2350		Вестник Войны	7533		Призрачный Танцор	1136		Повелитель Бури Фео	5620
13	Разбойник	2228		Искатель Сокровищ	7462		Жрец Евы	1019		Страж Теней Эура	5608
14	Разведчик	2128		Странник Бездны	6372		Деспот	980		Мастер Стихий Веньо	4901
15	Светлый Маг	2092		Властитель Огня	6141		Снайпер	956		Жрец Шилен Альгиза	4877
16	Светлый Рыцарь	1795		Мудрец Евы	5965		Авантюрист	893		Каратель Тира	4863
17	Рыцарь	1769		Огнелюб	5949		Страж Теней	799		Жрец Евы Альгиза	4773
18	Клерик	1557		Стрелок	5888		Аватар	793		Кардинал Альгиза	4719
19	Монах	1072		Призрачный Рейнджер	5163		Пожиратель Душ	790			4269
20	Оракул Евы	1030		Епископ	5092		Кардинал	785		Снайпер Эура	3703
21	Оракул Шилен	939		Рыцарь Шилен	5035		Призрачный Охотник	756		Виртуоз Иса	3503
22				Проповедник	4842		Дуэлист	754		Полководец Тира	3292
23				Палач	4788		Жрец Шилен	665		Храмовник Шилен Сигеля	3218
24				Паладин	4757		Полководец	645		Авантюрист Одала	3189
25				Мудрец Шилен	4657		Магистр Магии	576		Диверсант Эура	3098
26				Певец Заклинаний	4465		Чернокнижник	547		Пожиратель Душ Фео	3062
27				Арбалетчик	4212		Страж Лунного Света	531		Дуэлист Тира	3023
28				Серебряный Рейнджер	4129		Рыцарь Феикса	530		Инквизитор Фео	2926
29				Палач	4097		Диверсант	511		Деспот Иса	2787
30				Некоомант	4013		Виртуоз	481		Страж Лунного Света Эура	2535

Рисунок 2.11. - пример вывода данных в Google Doc

3. Вывод

В данной лабораторной работе было получено большое количество знаний в программировании на языке Python 3. Были освоены такие библиотеки, как BeautifulSoup4, urllib и gspread. Также все исходные данные можно найти на репозитории: <https://github.com/Zenkin/12onParsing>.