Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Образовательная программа Прикладная математика и информатика

бакалавриат

**01.03.02 Прикладная математика и информатика**

**О Т Ч Е Т**

**по учебной практике**

*Сбор данных о поступающих в ВУЗы России*

Выполнил студент группы 175

Булгаков Сергей Валерьевич

(ФИО)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*(подпись)*

**Проверили:**

М.Н.С. Паринов Андрей Андреевич (Международная научно-учебная лаборатория интеллектуальных систем и структурного анализа НИУ ВШЭ)

*(должность, ФИО руководителя от организации/НИУ ВШЭ)*

*\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**2017/2018 уч.г.**

Оглавление

[Введение 3](#_Toc525339998)

[Этапы выполнения задачи 3](#_Toc525339999)

[Реализация 3](#_Toc525340000)

[Общий код 3](#_Toc525340001)

[Бакалавриат СПбГУ (файл spbu\_bach.py) 4](#_Toc525340002)

[Магистратура СПбГУ (файл spbu\_mag.py) 5](#_Toc525340003)

[Бакалавриат РАНХиГС (файл ranepa\_bach.py) 5](#_Toc525340004)

[Магистратура РАНХиГС (файл ranepa\_mag.py) 6](#_Toc525340005)

[Магистратура МГУ (файл msu\_mag.py) 7](#_Toc525340006)

[Магистратура РГГУ (файл rggu\_mag.py) 7](#_Toc525340007)

[Примеры запуска программы 8](#_Toc525340008)

[Заключение 8](#_Toc525340009)

[Источники 9](#_Toc525340010)

# Введение

В Международной научно-учебной лаборатории интеллектуальных систем и структурного анализа НИУ ВШЭ стоит задача предсказания итогового ВУЗа для каждого абитуриента. Соответственно мне в качестве летней учебной практики была поставлена задача написания программы на языке Python 3, позволяющей осуществлять сбор списка поступающих в бакалавриат с сайтов СПбГУ и РАНХиГС.

# Этапы выполнения задачи

Для решения задачи было выполнено несколько подзадач:

* Исследование кода сайтов со списками и определение:
  + Информации, которую можно получить;
  + Алгоритма для считывания этой информации.
* Реализация алгоритмов;
* Запуск, тестирование, отладка и сохранение полученных данных;
* Публикация исходного кода и результирующих файлов.

В ходе выполнения задачи была написана документация к программе, а также настроен сервер непрерывной интеграции Travis CI для автоматического тестирования программы на удаленной машине. Дополнительно реализованы сборщики данных для магистратур СПбГУ, РАНХиГС, МГУ и РГГУ (будем называть ВУЗы, с которых программа умеет собирать дынные, отслеживаемыми).

# Реализация

Перейдем к обзору реализации задачи.

## Общий код

Несколько методов использованы практически в каждом ВУЗе, поэтому образованы следующие файлы.

* Работа с Интернетом (файл common\_html.py):
  + Методы для скачивания страницы из Интернета (если какая-то страница не отвечает, обращение к ней происходит до трех раз).
  + Методы для работы с кодом Интернет-страницы.
* Логирование (файл common\_logging.py):
  + Методы для логирования в файл lastLog.txt и в консоль.
* Многопоточное выполнение (файл common\_task\_queue.py):
  + Объект очереди задач, позволяющий добавлять задачи и ждать, пока все они не будут выполнены. Реализует шаблон проектирования «Производитель и потребители».
* Работа с JSON (файл common\_json.py):
  + Методы для сохранения результирующих файлов (имеют сигнатуру ({ссылка: список\_поступающих\_по\_ссылке}, название\_вуза)).
* Параметры сохранения JSON файлов (файл common\_properties.py):
  + Именованные константы.

## Бакалавриат СПбГУ (файл spbu\_bach.py)

Списки поступающих находятся на странице [4], где у каждого поступающего есть ссылки на таблицы с конкурсами.

На вход скрипта принимается:

* список страниц, с которых нужно собрать данные (None вместо списка, если нужно собрать данные со всевозможных конкурсов);
* список методов сохранения данных (если список пустой, выдается предупреждение).

Работа скрипта по шагам:

* Только если нужно построить список конкурсов:
  + Строится список поступающих.
  + Вытаскиваются конкурсы для каждого поступающего (асинхронно многопоточно).
  + Списки конкурсов объединяются в единый список.
* В словарь собираются поступающие для каждого конкурса (асинхронно многопоточно): парсятся общие для конкурса свойства, далее происходит итерация по таблице.
* Словарь передается в каждый метод сохранения с именем spbu-bach.

Следующие данные собираются с сайта о каждом поступающем (если есть):

* Образовательная программа
* Направление подготовки / Специальность
* Форма обучения (очная или заочная?)
* Основа обучения (бюджет или коммерция)
* ФИО абитуриента
* Дата рождения
* Тип отбора (общий конкурс или без вступительных испытаний или вне конкурса?)
* Сумма баллов (за конкурсы и за ИД)
* Сумма баллов (только за конкурсы)
* Баллы за конкурсы
* Баллы за индивидуальные достижения (сумма)
* Аттестат (подлинник или копия?)

## Магистратура СПбГУ (файл spbu\_mag.py)

Списки конкурсов находятся на странице [5].

На вход скрипта принимается:

* список страниц, с которых нужно собрать данные (None вместо списка, если нужно собрать данные со всевозможных конкурсов);
* список методов сохранения данных (если список пустой, выдается предупреждение).

Работа скрипта по шагам:

* Если нужно построить список конкурсов, то происходит построение.
* В словарь собираются поступающие для каждого конкурса (асинхронно многопоточно): парсятся общие для конкурса свойства, далее происходит итерация по таблице.
* Словарь передается в каждый метод сохранения с именем spbu-mag.

Следующие данные собираются с сайта о каждом поступающем (если есть):

* Направление подготовки / Специальность
* Образовательная программа
* ФИО абитуриента
* Тип отбора
* Баллы за конкурсы
* Баллы за индивидуальные достижения (сумма)
* Аттестат (подлинник или копия?)

## Бакалавриат РАНХиГС (файл ranepa\_bach.py)

Списки филиалов находятся на странице [6].

На вход скрипта принимается:

* список филиалов, с которых нужно собрать данные (None вместо списка, если нужно собрать данные со всевозможных филиалов);
* список методов сохранения данных (если список пустой, выдается предупреждение).

Работа скрипта по шагам:

* Идет спуск (внутри каждого пункта данные собираются асинхронно многопоточно):
  + Филиалы.
  + Направления.
  + Формы.
  + Образовательные программы.
  + Поступающие.
* Поступающие собираются в словарь.
* Словарь передается в каждый метод сохранения с именем ranepa-bach.

Следующие данные собираются с сайта о каждом поступающем (если есть):

* Признак
* Филиал
* Направление подготовки / Специальность
* Форма обучения (очная или заочная?)
* Образовательная программа
* Основа обучения (бюджет или коммерция)
* ФИО абитуриента
* Тип отбора (общий конкурс или без вступительных испытаний или вне конкурса?)
* Сумма баллов (за конкурсы и за ИД)
* Баллы за конкурсы
* Баллы за индивидуальные достижения (сумма)
* Аттестат (подлинник или копия?)

## Магистратура РАНХиГС (файл ranepa\_mag.py)

Списки филиалов находятся на странице [7].

На вход скрипта принимается:

* список филиалов, с которых нужно собрать данные (None вместо списка, если нужно собрать данные со всевозможных филиалов);
* список методов сохранения данных (если список пустой, выдается предупреждение).

Работа скрипта по шагам:

* Идет спуск (внутри каждого пункта данные собираются асинхронно многопоточно):
  + Филиалы.
  + Направления.
  + Формы.
  + Образовательные программы.
  + Поступающие.
* Поступающие собираются в словарь.
* Словарь передается в каждый метод сохранения с именем ranepa-mag.

Следующие данные собираются с сайта о каждом поступающем (если есть):

* Филиал
* Направление подготовки / Специальность
* Форма обучения (очная или заочная?)
* Образовательная программа
* Основа обучения (бюджет или коммерция)
* ФИО абитуриента
* Тип отбора (общий конкурс или без вступительных испытаний или вне конкурса?)
* Сумма баллов (за конкурсы и за ИД)
* Баллы за конкурсы
* Баллы за индивидуальные достижения (сумма)
* Аттестат (подлинник или копия?)

## Магистратура МГУ (файл msu\_mag.py)

Списки факультетов находятся на странице [8].

На вход скрипта принимается список методов сохранения данных (если список пустой, выдается предупреждение).

Работа скрипта по шагам:

* Собирается список факультетов.
* Собирается список поступающих на каждый факультет (асинхронно многопоточно).
* Поступающие собираются в словарь.
* Словарь передается в каждый метод сохранения с именем msu-mag.

Следующие данные собираются с сайта о каждом поступающем:

* Признак
* Факультет
* Направление подготовки / Специальность
* Форма обучения (очная или заочная?)
* ФИО абитуриента

## Магистратура РГГУ (файл rggu\_mag.py)

Списки направлений находятся на страницах [9] и [10].

На вход скрипта принимается список методов сохранения данных (если список пустой, выдается предупреждение).

Работа скрипта по шагам (для каждой ссылки):

* Собирается список конкурсов.
* Собирается список поступающих на каждом конкурсе (асинхронно многопоточно).
* Поступающие собираются в словарь.

Словари двух ссылок объединяются и передаются в каждый метод сохранения с именем rggu-mag.

Следующие данные собираются с сайта о каждом поступающем:

* Основа обучения (бюджет или коммерция)
* Направление подготовки / Специальность
* Форма обучения (очная или заочная?)

Также прикрепляются сырые данные из сайта.

# Примеры запуска программы

* В sample\_launcher.py показаны возможные варианты запуска сбора данных о поступающих для каждого отслеживаемого ВУЗа.
* sample\_launcher\_all\_mag.py запускает сбор данных о поступающих в магистратуру отслеживаемых ВУЗов (время выполнения порядка 10 минут).
* sample\_launcher\_all\_bach.py запускает сбор данных о поступающих в бакалавриат отслеживаемых ВУЗов (время выполнения порядка 40 минут).
* sample\_launcher\_all\_bach\_precomputed.py запускает сбор данных о поступающих в бакалавриат отслеживаемых ВУЗов (для СПбГУ используется заранее составленный список конкурсов; время выполнения порядка 20 минут).

# Заключение

Задача практики выполнена. Код проекта доступен на [1], списки поступающих доступны на [2] (общее число найденных записей о поступающих – примерно 200 000).

В ходе реализации задачи было приобретены и развиты следующие навыки:

* Программирование на языке Python 3:
  + Использование стандартных библиотек;
  + Нахождение и использование сторонней библиотеки (Beautiful Soup 4, доступна на [3]);
  + Шаблон проектирования «Производитель и потребители».
* Публикация исходного кода проекта на платформе GitHub:
  + Использование Git;
  + Создание GitHub релиза.
* Настройка сервера непрерывной интеграции Travis CI:
  + Добавление в проект файла конфигурации Travis CI;
  + Добавление в документацию проекта динамической иконки, отображающей статус успешности последней сборки на удаленном сервере.

Обсуждение продолжения проектной деятельности вместе с научным руководителем ведется.

# Источники

1. Исходный код проекта – <https://github.com/SerVB/abit-analytics>.
2. Страница с релизами таблиц поступающих – <https://github.com/SerVB/abit-analytics/releases>.
3. Сторонняя библиотека Beautiful Soup 4 – <https://pypi.org/project/beautifulsoup4/>.
4. Списки поступающих в бакалавриат СПбГУ – <https://cabinet.spbu.ru/Lists/1k_EntryLists/>.
5. Списки поступающих в магистратуру СПбГУ – <https://cabinet.spbu.ru/Lists/Mag_EntryLists/>.
6. Списки поступающих в бакалавриат РАНХиГС – <https://lk.ranepa.ru/pk/list.php?FT=1&FL=0>.
7. Списки поступающих в магистратуру РАНХиГС – <https://lk.ranepa.ru/pk/list.php?FT=1&FL=1>.
8. Списки поступающих в магистратуру МГУ – <http://cpk.msu.ru/daily>.
9. Списки поступающих в магистратуру РГГУ (бюджет) – <http://apply.rggu.ru/magistratura_list_budget/>.
10. Списки поступающих в магистратуру РГГУ (контракт) – <http://apply.rggu.ru/magistratura_list_contract/>.