

# HW #01: Python CLI Application

**Deadline: 27.11.2020, 08:00 (GMT+3)**

---

1. Описание задания	<b>2</b>
2. Критерии оценивания	<b>2</b>
3. Задача: Inverted Index	<b>3</b>
4. Общие рекомендации	<b>5</b>
5. Сроки сдачи и правила оформления задания	<b>5</b>

---



## 1. Описание задания

В данном ДЗ нужно написать приложение на Python, которое предоставляет консольный интерфейс для:

- построения инвертированного индекса и эффективного сохранения его на диске с помощью модуля [struct](#)<sup>1</sup>;
- поиска по инвертированному индексу в кодировках utf-8 и cp1251.

## 2. Критерии оценивания

Балл за задачу складывается из:

- **30%** - правильная реализация поиска
- **40%** - эффективность сжатия индекса, более точная формула:
  - $40\% \times \min(1.5, 23^2 \text{ MiB} / \text{compression\_size})$
- **20%** - качество покрытия тестами, точная формула:
  - $20\% \times \min([\text{test\_coverage} / 0.8], 1.0)$
- **10%** - качество кода (pylint<sup>3</sup>), точная формула:
  - $10\% \times \min([\text{lint\_quality} / 8.0], 1.0)$

Бонусы и штрафы:

- **100%** за плагиат в решениях (всем участникам процесса)
- **100%** за использование pickle, zlib и других библиотек вместо struct
- **30%** за посылку решения в течение недели после deadline (**-100%** спустя неделю)
- ~~**5%** за каждую повторную посылку<sup>4</sup>~~
- до **20%** за эффективность сжатия<sup>5</sup>

---

<sup>1</sup> Да, это реальная жизнь, вам придется выйти немного за рамки того, что изучали. Все что нужно - выбрать правильную кодировку и пользоваться методами pack, unpack и calcsz.

<sup>2</sup> Приблизительный размер сжатого индекса для датасета Wikipedia (sample)

<sup>3</sup> pylint версии 2.6.0

<sup>4</sup> Поскольку задание первое, то в целях знакомства разрешаем сделать больше посылок без штрафа

<sup>5</sup> Баллы свыше 100% можно использовать для повышения финальной оценки по курсу. Для этого необходимо написать в конце курса письмо о запросе их учета.

### 3. Задача: Inverted Index

Приложение (`inverted_index.py`) должно предоставлять следующий CLI<sup>6</sup>:

1. Построение инвертированного индекса на основе датасета (см. формат ниже) и его эффективное сжатие для сохранения на диск:

```
$ python3 inverted_index.py build \  
    --dataset /path/to/dataset \  
    --output /path/to/inverted.index
```

2. Реализация поиска со следующим консольным интерфейсом:

```
$ python3 inverted_index.py query --index /path/to/inverted.index \  
    --query-file-utf8 /path/to/quries.txt
```

```
$ cat /path/to/quries.txt | python3 inverted_index.py query \  
    --index /path/to/inverted.index \  
    --query-file-utf8 -
```

```
$ python3 inverted_index.py query --index /path/to/inverted.index \  
    --query-file-cp1251 /path/to/quries.txt
```

```
$ cat /path/to/quries.txt | python3 inverted_index.py query \  
    --index /path/to/inverted.index \  
    --query-file-cp1251 -
```

```
$ python3 inverted_index.py query \  
    --index /path/to/inverted.index \  
    --query first query [--query the second query]7
```

---

<sup>6</sup> Должны поддерживаться как абсолютные, так и относительные пути

<sup>7</sup> Консоль будет работать в кодировке utf-8



Чтение сжатого инвертированного индекса с диска в память и “обстрел” запросами, которые предоставляются файлом в кодировке utf-8 или cp1251. В каждой строке находится ровно один запрос, состоящий из любого числа слов, разделенных пробельными символами. Выходной формат вывода см. ниже.

Одновременный вызов `--query-file-cp1251` и `--query-file-utf8` должен приводить к ошибке (return code приложения не должен быть равен 0).

## Входные данные

Wikipedia (sample):

- доступен по ссылке для скачивания - [здесь](#);
- предполагается, что файл доступен в режиме read-only в локальной директории проекта под названием `wikipedia_sample.txt`;
- формат: текст в кодировке utf-8
- в каждой строке:

```
article_ID(int) <tab> article_name <spaces> article_content
```

В качестве уникальных термов (слов) для поиска для однозначности проверки решения используем конструкцию:

```
content = article_name + " " + article_content
words = content.split()
```

Пример:

```
12    Anarchism           Anarchism is often defined as a ...
```

## Выходной формат “обстрела”

По результатам “обстрела” stdout должен содержать **только** ответы на запросы (всю остальную вспомогательную информацию пишите в stderr или в логи). Ответ на запрос - список идентификаторов документов (статей Википедии), разделенных запятыми. Пример:

- запрос в файле: “long query”, состоит из двух слов “long” и “query”
- допустим в датасете только 3 документа 151, 13, 3998 содержат **одновременно оба** этих слова, тогда ваш ответ: “151,13,3998”. Порядок предоставленных документов в ответе не важен (может быть любым). Но

проверяется, что Вы нашли абсолютно все нужные документы и ничего лишнего.

## 4. Общие рекомендации

При решении задач старайтесь следовать следующим рекомендациям:

- держите уровень покрытия кода тестами на уровне 80+%, следуйте TDD (сначала тесты, потом реализация);
- отделяйте фазу рефакторинга от фазы добавления новой функциональности, т.е.
  - фиксируем функциональность, все тесты зеленые;
  - проводим рефакторинг;
  - по окончании фазы рефакторинга снова все тесты зеленые;
- следите за скоростью выполнения unit-test'ов, несколько секунд - это хорошо, в противном случае нужно уменьшать размер тестируемых датасетов или разделять тесты на фазы (см. видео про `mark.slow`);
- следите за качеством кода и проверяйте "глупые" ошибки с помощью `pylint`, следите за поддерживаемостью и читаемостью кода;

## 5. Сроки сдачи и правила оформления задания

Оформление задания:

- Выполненное ДЗ запакуйте в архив **MADEPY20Q4\_<Surname>\_<Name>\_HW1.zip**, например, для Алексея Драля -- **MADEPY20Q4\_Dral\_Alexey\_HW1.zip**. Если ваше решение лежит в папке `my_solution_folder`, то для создания архива `hw.zip` на Linux и Mac OS выполните команду:
  - `zip -r hw.zip my_solution_folder/*`
- На Windows 7/8/10: необходимо выделить необходимое для сдачи содержимое директории `my_solution_folder/` нажать правую кнопку мыши на одном из выделенных объектов, выбрать в открывшемся меню "Отправить >", затем "Сжатая ZIP-папка". Теперь можно переименовать архив.
- Перед проверкой убедитесь, что дерево вашего архива выглядит так (в том числе не должно быть лишних файлов и директорий):
  - | **MADEPY20Q4\_<Surname>\_<Name>\_HW1.zip**



- | ---- task\_<Surname>\_<Name>\_inverted\_index.py
- | ---- test\_<Surname>\_<Name>\_inverted\_index.py<sup>8</sup>
- При несовпадении дерева вашего архива с представленным деревом ваше решение не будет возможным автоматически проверить, а значит, и оценить его.
- Для того, чтобы сдать задание необходимо:
  - Зарегистрироваться и залогиниться в сервисе [Everest](#)
  - Перейти на страницу приложения ["BigData Team | MADE Python Grader"](#)
  - Выбрать вкладку Submit Job (если отображается иная).
  - Выбрать в качестве "Task" значение: "HW1: Inverted index"<sup>9</sup>
  - Загрузить в качестве "Task solution" архив с решением
  - В качестве Sender ID указать свой индивидуальный id слушателя (если после объявления о том, что всем id разосланы, вы свой не получили, свяжитесь с нами).

Любые вопросы / комментарии / предложения:

- по работе тестирующей системы просьба писать на почту [grader@bigdatateam.org](mailto:grader@bigdatateam.org);
- по заданиям и в целом по курсу - в [Discord-канал](#) курса (#python). [Приглашение на сервер](#), если еще не успели присоединиться.

---

<sup>8</sup> Тесты вашего приложения, которые можно запустить с помощью "PYTHONPATH=. pytest test\_....py".

<sup>9</sup> Сервисный ID: python.inverted\_index