

МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ»
ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра Информационных Систем

Пояснительная записка

К курсовой работе
по дисциплине «Алгоритмы и структуры данных»

Тема: «Сортировка и поиск данных»

Студент гр. 8894		Кривенкин В.П.
Проверил		Молдовян Д.Н.

Санкт-Петербург
2020

Содержательная постановка задачи

Разработать информационно-поисковую систему, позволяющую осуществлять поиск, сортировку данных, изменять, добавлять и удалять данные в процессе работы программы.

Обработка библиотечной информации (isbn книги, название книги, имя автора, издательство, количество страниц).

При реализации программы необходимо использовать алгоритм пирамидальной сортировки.

Исходные данные вводятся как с клавиатуры, так и из текстового файла. Файл с данными создается самостоятельно. Требуется по заданным ключам произвести сортировку данных. Реализовать поиск данных по задаваемому значению ключа.

Анализ решения задачи

Информационно-поисковая система Cheryl реализована на языке программирования высокого уровня Python и представляет собой консольное приложение, позволяющее пользователю управлять базой книг, хранящейся на жестком диске.

После запуска программа ожидает следующую команду от пользователя. Список доступных команд приведен ниже:

- print — показать все имеющиеся в базе книги;
- sort — сортировать книги по одному из ключей;
- find — найти книгу по ключу;
- add — добавить новую книгу;
- delete — удалить книгу;
- change — изменить книгу;
- quit — выйти из программы;
- help — отобразить доступные команды и их описания;

Во время запуска программы Cheryl в качестве аргумента может быть передано имя файла, в котором хранятся данные о книгах. Если этого не сделано, то будет создан новый файл с именем my-books.csv, в который будет записываться информация о книгах.

Программа Cheryl поддерживает механизм генерации случайных книг. Чтобы создать базу, содержащую нужное количество книг, необходимо запустить скрипт generate.py, передав ему в качестве аргумента количество книг. После этого будет создан файл my-books.csv, содержащий нужное количество случайно сгенерированных книг.

Спецификация программы

Входные данные

На вход программе может быть передан файл формата csv (comma-separated values), содержащий информацию о книгах. Строка такого файла должна выглядеть следующим образом:

'000-000-001','Jolly Massive Theory','Elliot Wright','UK Press','210' .

В этой строке через запятую в одинарных кавычках перечислены данные о книге: isbn, название книги, имя автора, издательство и количество страниц.

База книг хранится на диске как последовательность таких строк, сохраненных в файле.

Выходные данные

После завершения программы данные о книгах сохраняются в файл формата csv (см. пункт «Входные данные» выше) с учетом всех изменений внесенных пользователем.

Сценарий работы программы

Продemonстрируем, как работает программа.

С помощью скрипта `generate.py` создадим базу из пяти книг:

```
valery@dave: ~/Desktop/cheryl/cheryl
File Edit View Search Terminal Help
(venv) valery@dave:~/Desktop/cheryl/cheryl$ ls
cheryl      generate.py  LICENSE     README.md   test
cheryl.py   generator   note.docx   requirements
(venv) valery@dave:~/Desktop/cheryl/cheryl$ ./generate.py 5
(venv) valery@dave:~/Desktop/cheryl/cheryl$ cat my-books.csv
'000-000-000','Clean Scrawny Architects','Floyd Yeabsley','NH Press','83'
'000-000-001','Clean Lazy Children','Earl Tomson','LB Press','865'
'000-000-002','Glamorous Massive Methods','Ben Pierce','SY Press','266'
'000-000-003','Elegant Bewildered Laws','Andrew Roope','UU Press','242'
'000-000-004','Jolly Holographic Soldiers','Howard Rudd','FW Press','470'
(venv) valery@dave:~/Desktop/cheryl/cheryl$ |
```

Мы видим, что был создан файл `my-books.csv`, содержащий информацию о пяти книгах. Теперь запустим саму программу Cheryl, передав в качестве аргумента имя только что созданного файла:

```
valery@dave: ~/Desktop/cheryl/cheryl
File Edit View Search Terminal Help
(venv) valery@dave:~/Desktop/cheryl/cheryl$ ./cheryl.py -db my-books.csv
Cheryl version 0.1
Enter '.help' for usage hints.
cheryl> |
```

Мы успешно запустили программу и видим строку, ожидающую очередную команду. Введем команду «help», чтобы увидеть информацию о доступных командах:

```
valery@dave: ~/Desktop/cheryl/cheryl
File Edit View Search Terminal Help
(venv) valery@dave:~/Desktop/cheryl/cheryl$ ./cheryl.py -db my-books.csv
Cheryl version 0.1
Enter '.help' for usage hints.
cheryl> help
    print          Print stored books
    sort           Sort books by key
    find           Find a book by key
    add            Add a new book
    delete         Delete a book by key
    change         Change book attribute by key
    quit           Exit program
    help           Print information about commands
cheryl> |
```

Теперь у нас есть некоторое представление о том, как пользоваться программой Cheryl. Попробуем отобразить все имеющиеся книги с помощью команды «print»:

```
valery@dave: ~/Desktop/cheryl/cheryl
File Edit View Search Terminal Help
(venv) valery@dave:~/Desktop/cheryl/cheryl$ ./cheryl.py -db my-books.csv
Cheryl version 0.1
Enter '.help' for usage hints.
cheryl> help
    print          Print stored books
    sort           Sort books by key
    find           Find a book by key
    add            Add a new book
    delete         Delete a book by key
    change         Change book attribute by key
    quit           Exit program
    help           Print information about commands
cheryl> print
There is no such command 'print'. Please, try again.
cheryl> |
```

Произошла досадная опечатка и мы видим на экране сообщение об этом.

Попробуем команду «print» снова:

```
cheryl> print
| 000-000-000 | Clean Scrawny Architects | Floyd Yeabsley | NH Press | 83 |
| 000-000-001 | Clean Lazy Children | Earl Tomson | LB Press | 865 |
| 000-000-002 | Glamorous Massive Methods | Ben Pierce | SY Press | 266 |
| 000-000-003 | Elegant Bewildered Laws | Andrew Roope | UU Press | 242 |
| 000-000-004 | Jolly Holographic Soldiers | Howard Rudd | FW Press | 470 |
cheryl> |
```

На этот раз все в порядке и мы видим информацию о книгах в табличной форме. Теперь отсортируем книги по названию (title) при помощи команды «sort» и снова отобразим их:

```
cheryl> sort
  sort by key> title
Books have been sorted by title
cheryl> print
| 000-000-001 | Clean Lazy Children | Earl Tomson | LB Press | 865 |
| 000-000-000 | Clean Scrawny Architects | Floyd Yeabsley | NH Press | 83 |
| 000-000-003 | Elegant Bewildered Laws | Andrew Roope | UU Press | 242 |
| 000-000-002 | Glamorous Massive Methods | Ben Pierce | SY Press | 266 |
| 000-000-004 | Jolly Holographic Soldiers | Howard Rudd | FW Press | 470 |
cheryl> |
```

Теперь найдем книгу по ключу isbn при помощи команды find:

```
cheryl> find
  find by key> isbn
  isbn> 000-000-003
| 000-000-003 | Elegant Bewildered Laws | Andrew Roope | UU Press | 242 |
cheryl> |
```

Книга успешно найдена. Искать, изменять и удалять книги можно не только по isbn, но и по названию. Теперь попробуем добавить книгу:

```
cheryl> add
  isbn> 000-000-555
  title> Catch-22
  author>
    author cannot be empty
  author> Joseph Heller
  publisher> Pilot Press
  pages> 442
'Catch-22' by Joseph Heller has been successfully added
```

Книга была успешно добавлена.

Удалим книгу с названием «Clean Lazy Children»:

```
cheryl> delete
  find by key> title
  title> clean lazy children
'Clean Lazy Children' by Earl Tomson has been successfully deleted
cheryl> print
| 000-000-555 |          Catch-22          | Joseph Heller | Pilot Press | 442
| 000-000-000 | Clean Scrawny Architects | Floyd Yeabsley | NH Press   | 83
| 000-000-003 | Elegant Bewildered Laws  | Andrew Roope  | UU Press   | 242
| 000-000-002 | Glamorous Massive Methods | Ben Pierce    | SY Press   | 266
| 000-000-004 | Jolly Holographic Soldiers | Howard Rudd   | FW Press   | 470
cheryl> |
```

Теперь попробуем изменить количество страниц у книги «Clean Scrawny Architects» на 120 вместо 83:

```
cheryl> change
  find by key> title
  title> clean scrawny architects
  update key> pages
  new pages> 120
pages has been successfully changed to '120'
cheryl> print
| 000-000-555 |          Catch-22          | Joseph Heller | Pilot Press | 442
| 000-000-000 | Clean Scrawny Architects | Floyd Yeabsley | NH Press   | 120
| 000-000-003 | Elegant Bewildered Laws  | Andrew Roope  | UU Press   | 242
| 000-000-002 | Glamorous Massive Methods | Ben Pierce    | SY Press   | 266
| 000-000-004 | Jolly Holographic Soldiers | Howard Rudd   | FW Press   | 470
cheryl> |
```

Отсортируем книги по количеству страниц и проверим, сохранятся ли данные на диске после выхода из программы:

```
cheryl> sort
  sort by key> pages
Books have been sorted by pages
cheryl> quit
Bye.
(venv) valery@dave:~/Desktop/cheryl/cheryl$ cat my-books.csv
'000-000-000','Clean Scrawny Architects','Floyd Yeabsley','NH Press','120'
'000-000-003','Elegant Bewildered Laws','Andrew Roope','UU Press','242'
'000-000-002','Glamorous Massive Methods','Ben Pierce','SY Press','266'
'000-000-555','Catch-22','Joseph Heller','Pilot Press','442'
'000-000-004','Jolly Holographic Soldiers','Howard Rudd','FW Press','470'
(venv) valery@dave:~/Desktop/cheryl/cheryl$ |
```

Приведенный выше сценарий работы программы демонстрирует основные ее возможности.

Исходный код

Исходный код информационно-поисковой системы Cheryl доступен по ссылке: <https://github.com/valery42/cheryl>

Структура программы

Структурно информационно-поисковая система Cheryl состоит из двух классов: Engine и Handler и ряда вспомогательных функций.

Класс Engine — это движок информационно-поисковой системы. В его задачи входит загрузка и сохранение базы книг, низкоуровневые операции добавления, удаления, изменения и сортировки книг. Кроме того этот класс хранит некоторую дополнительную информацию для того, чтобы правильно отображать книги и не сортировать их лишней раз, когда они уже отсортированы по одному из ключей. Класс содержится в модуле engine.py и доступен по ссылке: <https://github.com/valery42/cheryl/blob/master/cheryl/engine.py>

Класс Handler гораздо «умнее» класса Engine, он оборачивает его методы для интерактивного взаимодействия с пользователем. С помощью ряда вспомогательных функций этот класс постоянно проверяет информацию, полученную от пользователя на правильность и при необходимости сообщает пользователю об ошибках. Класс содержится в модуле handler.py и доступен по ссылке: <https://github.com/valery42/cheryl/blob/master/cheryl/handler.py>

Вспомогательные функции, используемые классом Handler содержатся в модулях checkers.py, converters.py и utils.py. Модуль checkers.py содержит функции, проверяющие количество страниц и isbn на корректность. Модуль converters.py содержит функции, осуществляющие конвертацию записи на диске в книгу в памяти компьютера и обратно. Модуль utils.py содержит ряд функций для получения сообщений для пользователя, создания новых книг и отображения книг на экране. Ссылка: <https://github.com/valery42/cheryl/blob/master/cheryl/utils.py>

Алгоритмы

Алгоритм сортировки

В качестве алгоритма сортировки используется heapsort (алгоритм пирамидальной сортировки). Вычислительная сложность этого алгоритма в худшем, среднем и лучшем случае равна $O(n \log n)$. Сортировка может быть осуществлена по одному из ключей книги (isbn, название книги, имя автора, издательство, количество страниц). Реализация алгоритма содержится в модуле sort и доступна по ссылке:

<https://github.com/valery42/cheryl/blob/master/cheryl/sort.py>

Алгоритм реализован по псевдокоду из книги «Introduction to Algorithms» Томаса Кормена и др. Ссылка на книгу приведена в списке литературы.

Особенностью алгоритма пирамидальной сортировки является использование вспомогательной структуры данных, а именно кучи (max-heap), реализованной на массиве. Поэтому для успешной реализации алгоритма heapsort на языке высокого уровня необходимо написать две вспомогательные функции: max_heapify, вычислительная сложность которой $O(\log n)$ и build_max_heap, вычислительная сложность которой $O(n)$. Первая функция реализует свойство кучи (max-heap property), а вторая строит кучу из неупорядоченного массива данных. Подробнее об алгоритме см. книгу Кормена и др.

Алгоритм поиска

Для поиска книги по isbn или title используется алгоритм binary search (алгоритм бинарного поиска). Вычислительная сложность в худшем и среднем случае равна $O(\log n)$, в лучшем — $O(1)$. Реализация доступна по ссылке:

<https://github.com/valery42/cheryl/blob/master/cheryl/search.py>

Алгоритм реализован по псевдокоду с сайта wikipedia.org. Подробнее об алгоритме там же: https://en.wikipedia.org/wiki/Binary_search_algorithm

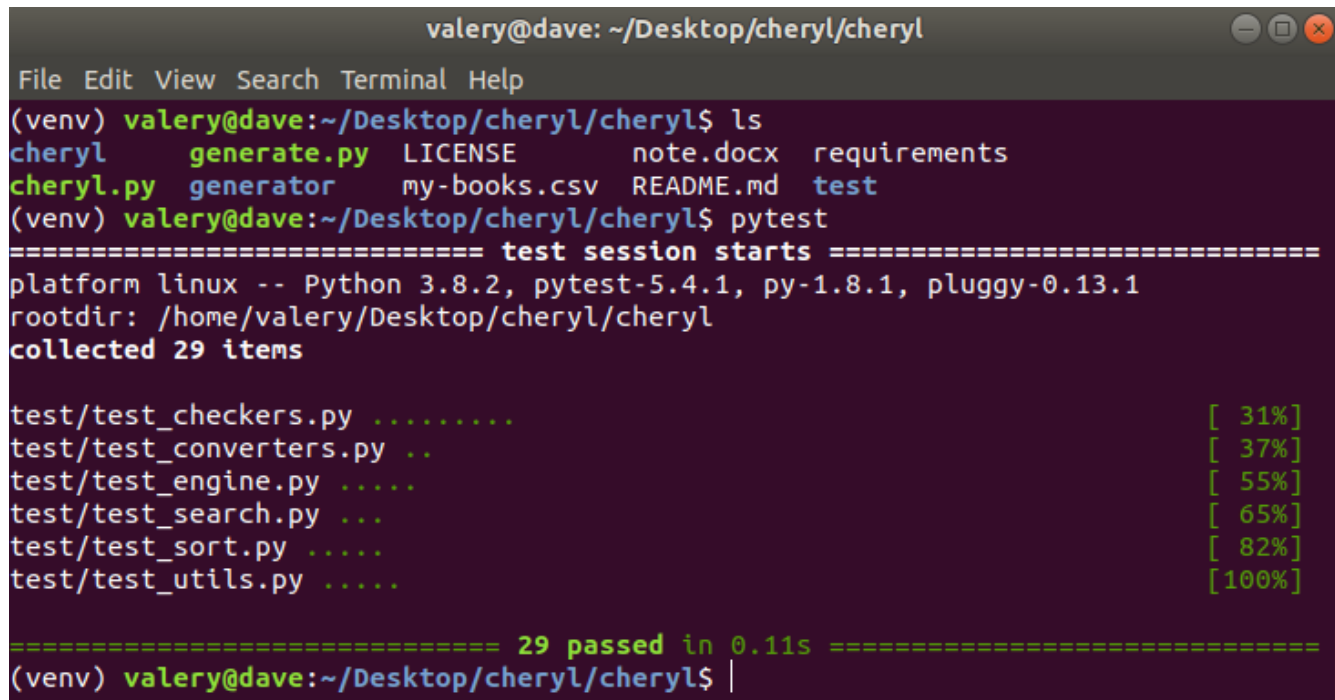
Тестирование программы

Единственной гарантией хоть какой-то правильности работы программы является наличие тестов. Для тестирования информационно-поисковой системы Cheryl используется фреймворк для тестирования Pytest. Он позволяет быстро писать тесты используя простые assert выражения, тестировать отдельные модули и даже отдельные функции, объединять тесты сходной функциональности в классы и многое другое.

Тесты для программы Cheryl содержатся в пакете test и доступны по ссылке:

<https://github.com/valery42/cheryl/tree/master/test>

Запустим все тесты и убедимся, что программа работает корректно:



```
valery@dave: ~/Desktop/cheryl/cheryl
File Edit View Search Terminal Help
(venv) valery@dave:~/Desktop/cheryl/cheryl$ ls
cheryl      generate.py  LICENSE      note.docx   requirements
cheryl.py   generator   my-books.csv README.md   test
(venv) valery@dave:~/Desktop/cheryl/cheryl$ pytest
===== test session starts =====
platform linux -- Python 3.8.2, pytest-5.4.1, py-1.8.1, pluggy-0.13.1
rootdir: /home/valery/Desktop/cheryl/cheryl
collected 29 items

test/test_checkers.py ..... [ 31%]
test/test_converters.py .. [ 37%]
test/test_engine.py ..... [ 55%]
test/test_search.py ... [ 65%]
test/test_sort.py ..... [ 82%]
test/test_utils.py ..... [100%]

===== 29 passed in 0.11s =====
(venv) valery@dave:~/Desktop/cheryl/cheryl$ |
```

Мы видим, что все тесты завершились успешно.

Анализ результатов и выводы

В ходе данной курсовой работы мною была реализована информационно-поисковая система Cheryl на языке программирования Python, которая позволяет пользователю управлять базой, содержащей информацию о книгах. Реализованная

система в высокой степени интерактивна и информативна: ни одна ошибка не пропадает в никуда, пользователь всегда о ней узнает и, скорее всего, не повторит ее.

Для построения программы были реализованы два алгоритма: алгоритм сортировки heapsort и алгоритм поиска binary search.

Все компоненты системы были тщательно протестированы как с помощью автоматических тестов, так и вручную.

Литература

1. Introduction to Algorithms (The MIT Press); Cormen, Leiserson, Rivest, Stein; The MIT Press; 3 edition (July 31, 2009)
2. Code Complete (Developer Best Practices); Steve McConnell; Microsoft Press; 2 edition (June 9, 2004)
3. Binary Search Algorithm: https://en.wikipedia.org/wiki/Binary_search_algorithm
4. Python Official Website: <https://www.python.org/>
5. Pytest Framework: <https://docs.pytest.org/en/latest/>