

**Кафедра «Информационных систем»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**по выполнению рейтинговой работы**

**по дисциплине «Алгоритмизация и программирование»**

Направление подготовки: *09.03.03 «Прикладная информатика»*

Уровень высшего образования:

*Бакалавриат*

**Москва 2019**

**СОДЕРЖАНИЕ**

[1. ОБЩИЕ ПОЛОЖЕНИЯ 3](#_gjdgxs)

[2. ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ РЕЙТИНГОВОЙ РАБОТЫ 4](#_30j0zll)

[3. РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РЕЙТИНГОВОЙ РАБОТЫ 8](#_1fob9te)

[4. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ РЕЙТИНГОВОЙ РАБОТЫ 9](#_3znysh7)

[6. СТРУКТУРА И СОДЕРЖАНИЕ РАСЧЕТНО-АНАЛИТИЧЕСКОГО ЗАДАНИЯ РЕЙТИНГОВОЙ РАБОТЫ 10](#_2et92p0)

[7. КРИТЕРИИ ОЦЕНКИ РЕЙТИНГОВОЙ РАБОТЫ 11](#_tyjcwt)

[8. ПРИЛОЖЕНИЕ 13](#_3dy6vkm)

# 1. ОБЩИЕ ПОЛОЖЕНИЯ

Рейтинговая работа по дисциплине выполняется обучающимся в ходе самостоятельной работы и является обязательным элементом балльно-рейтинговой системы (БРС) Университета.

**Рейтинговая работа – *Расчетно-аналитическое задание***

***Расчетно-аналитическое задание*** – самостоятельная работа, требующая от обучающегося умений применять полученные в ходе изучения дисциплины «Алгоритмизация и программирование» знания и навыки при решении алгоритмических задач и выполнении заданий с использованием типовых алгоритмических конструкций. Выполнение расчетно-аналитического задания ориентировано на развитие навыков программирования на языке Python, логического и аналитического мышления.

Рейтинговая работа в виде расчетно-аналитического задания состоит из задания, по которому необходимо написать программу на языке Python. При этом оценивается не только получение конечного результата, но и качество кода программы и следование стилю, изложенному в «Соглашении PEP 8» по написанию кода на Python. Решение задач расчетно-аналитического задания рассчитано на обретение навыков правильного применения языка Python в конкретных условиях, формирование общепрофессиональных компетенций ОПК-2, ОПК-3, ОПК-4, ОПК-5, ОПК-7.

Задание для рейтинговой работы соответствует любому количеству вариантов.

**Выбор варианта исходных данных рейтинговой работы осуществляется на основе выборки символов, составляющих Фамилию Имя Отчество обучающегося, с последующим переводом каждого символа в Unicode (Юникод).**

# 2. ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ РЕЙТИНГОВОЙ РАБОТЫ

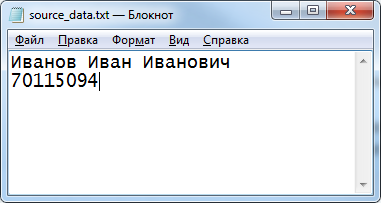
Дан ряд целых чисел, полученный в результате обработки символов Фамилии, Имени, Отчества обучающегося, а также его ID. Необходимо отсортировать набор данных двумя способами, а также выполнить некоторые арифметические расчеты.

***Исходные данные***

ФИО и ID подготавливаются заранее в текстовом файле с именем source\_data.txt в кодировке Unicode (Юникод). Каждое слово, составляющее ФИО должно быть с заглавной буквы. Слова между собой должны быть разделены пробелом. Точка в конце не ставится. Текстовый файл с исходными данными должен иметь следующий вид:

Иванов Иван Иванович

70115094



**Примечание:**

1. *ФИО – Иванов Иван Иванович, и ID – 70115094 представлены в качестве примера. Естественно, при формировании файла source\_data.txt, нужно вносить свои ФИО и ID.*
2. *ФИО и ID разделены символом enter (u000a). Данный символ в последующей обработке набора данных* ***не должен участвовать****. После номера ID символ enter не ставится; таким образом, в файле source\_data.txt должно быть две строки.*
3. ***Пробелы*** *между словами ФИО в обработке набора данных* ***не должны участвовать***.

**Требуется написать программу на языке Python, которая выполняет следующие задачи:**

1. Считать исходные данные из текстового файла source\_data.txt

2. Вычислить целое значение, получившееся в результате деления ID на количество символов, составляющих ФИО (количество символов подсчитывается **без учета пробелов**).

3. Определить направление сортировки в зависимости от числа, полученного в п.2:

- сортировка по возрастанию, если число чётное;

- сортировка по убыванию, если число нечётное.

4. Сформировать набор данных (список) из кодов Юникода **каждого символа ФИО** (исключая пробелы между словами), переведенных в десятичную форму.

5. Выполнить сортировку набора данных по убыванию или по возрастанию. **Сортировку выполнить двумя различными способами**, например: сортировка выбором и сортировка методом простого обмена (пузырька). В программном коде должны быть представлены комментарии, обозначающие тот или иной алгоритм сортировки.

# алгоритм сортировки по методу пузырька

# алгоритм "Шейкерная сортировка"

6. Вычислить среднее арифметическое значение набора данных. При необходимости, результат округлить до третьего знака после запятой.

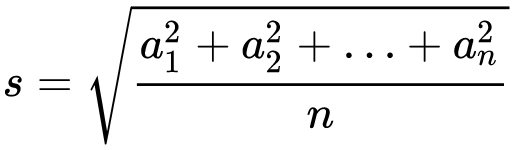
7. Вычислить среднее квадратическое значение набора данных. При необходимости, результат округлить до третьего знака после запятой.

**Обратите внимание на важные замечания:**

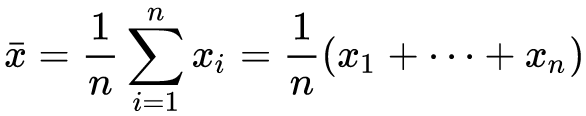
1. Не допускается использование готовых функций по сортировки набора данных, например: метод .sort(), функции sorted(), reverse() и т.п. Использование данных функций возможно лишь для самопроверки, на этапе тестирования результатов работы двух алгоритмов сортировки. В окончательном варианте программного кода этих функций и методов не должно быть.

2. Вычисление среднего арифметического и среднеквадратичного значения также должно быть выполнено без использования готовых функций, т.е. нужно написать алгоритм.

3. **Справочно**: среднее квадратическое (квадратичное) набора данных — число, равное квадратному корню из среднего арифметического квадратов данных чисел *a1, a2, …, an*:



4. **Справочно**: среднее арифметическое значение набора чисел — число, равное сумме всех чисел множества, делённое на их количество:



**Результат:**

Результатом правильной работы программы является:

1. текстовый файл с результатами расчетов result.txt. В консоль ничего не выводится. При перезапуске программы файл result.txt должен быть перезаписан.

Формат текстового файла результатов result.txt

1. Исходные данные: Иванов Иван Иванович; ID: 70115094

2. 3505754

3. Направление сортировки: по возрастанию, так как число 3505754 – чётное

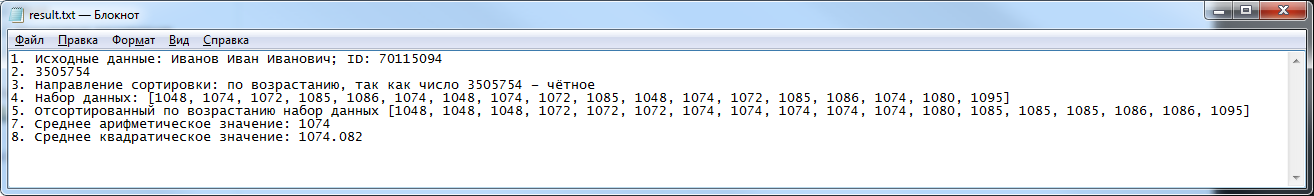
4. Набор данных: [1048, 1074, 1072, 1085, 1086, 1074, 1048, 1074, 1072, 1085, 1048, 1074, 1072, 1085, 1086, 1074, 1080, 1095]

5. Отсортированный по возрастанию набор данных [1048, 1048, 1048, 1072, 1072, 1072, 1074, 1074, 1074, 1074, 1074, 1080, 1085, 1085, 1085, 1086, 1086, 1095]

7. Среднее арифметическое значение: 1074

8. Среднее квадратическое значение: 1074.082

Данные залитые зеленой заливкой – это данные, которые формируются в зависимости от расчета по выполненной программе. Остальные данные – неизменяемые. Кодировка в текстовом файле должна быть Unicode. **Путь** к входному и выходному файлам (source\_data.txt и result.txt) в коде программы нужно сделать **относительным**.



# 3. РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РЕЙТИНГОВОЙ РАБОТЫ

Для подготовки к выполнению расчетно-аналитического задания рекомендуется самостоятельное изучение учебной литературы, использование справочной литературы и интернет - ресурсов. По мере изучения темы следует ознакомится с видеолекциями.

При этом оценивается:

1) правильность и обоснованность выбора алгоритмов сортировки;

2) умение работать с данными на внешних текстовых файлах;

3) умение использовать типовые алгоритмические конструкции

4) аккуратность, полнота выполнения расчетно-аналитического задания.

Программа должна быть составлена самостоятельно. Не самостоятельно выполненная программа лишает студента возможности проверить степень своей подготовленности по теме. Если преподаватель установит несамостоятельное выполнение программы, то она не будет зачтена.

Если работа не соответствует этим требованиям, то она возвращается студенту на доработку. Правильно выполненная рейтинговая работа наряду с выполнением тестовых заданий необходима для прохождения промежуточной аттестации по дисциплине «Алгоритмизация и программирование». Задания рейтинговой работы выполняются в течение семестра, в срок, определяемый графиком учебного процесса, до проведения промежуточной аттестации.

# 4. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ РЕЙТИНГОВОЙ РАБОТЫ

Рейтинговая работа выполняется в электронной форме и размещается обучающимся в личном кабинете в виде архивного файла. Состав архивного файла:

- расчетно-аналитическое задание с именем файла ФИО.docx

(пример: «Иванов АС.docx»)

- файл с исходными данными source\_data.txt

- файл с результатами сортировки и расчетами result.txt

- файл программы Ivanov\_AS.py

Имя размещаемого архивного файла должно содержать фамилию и инициалы студента, например: «Иванов\_АС.zip». Для создания архива можно воспользоваться свободно распространяемой программой 7-Zip.

Расчетно-аналитическое задание должно быть оформлено в виде документа Word, допускается вставка таблиц, скопированных из Excel. Текст на листе должен иметь книжную ориентацию, альбомная ориентация допускается только для таблиц и схем приложений. Поля страницы должны иметь следующие размеры: левое – 3 см, правое 1,5 см, верхнее – 2 см, нижнее – 2 см. Текст печатается через полтора интервала шрифтом Times New Roman, 14 кегль (для сносок 12 кегль), допускаются переносы в словах. Абзац – 1,25 см. Формулы необходимо представлять с помощью редактора формул.

Номера страниц размещаются в нижнем правом углу. Применяется сквозная нумерация листов, начиная с титульного листа и включая приложения (если есть). Номер листа на титульном листе не проставляют. Второй лист расчетно-аналитического задания – содержание. Решение задачи рейтинговой работы рассчитано на формирование навыков по работе со стандартными алгоритмическими конструкциями и работе с текстовыми файлами с использованием языка Python. Расчетно-аналитическое задание должно быть выполнено с использованием среды разработки PyCharm Edu / Community. Использование других сред разработки не рекомендуется.

# 6. СТРУКТУРА И СОДЕРЖАНИЕ РАСЧЕТНО-АНАЛИТИЧЕСКОГО ЗАДАНИЯ РЕЙТИНГОВОЙ РАБОТЫ

Расчетно-аналитическое задание рейтинговой работы должно содержать следующие структурные элементы:

- титульный лист (см. приложение);

- содержание;

- задание с исходными данными;

- описание последовательности выполнения работы;

- обоснование выбора того или иного алгоритма сортировки числового набора данных;

- листинг программного кода **с подробными комментариями** к каждому оператору;

- результаты выполнения программы, где нужно представить описание файла result.txt;

- список используемой литературы, к котором необходимо указать основные и дополнительные источники, использовавшиеся при выборе алгоритмов сортировки и написании программного кода. Также допускается указание специализированных Интернет-ресурсов, посвященных тематике Python. Список литературы следует оформлять по ГОСТ 7.05.2008.

# 7. КРИТЕРИИ ОЦЕНКИ РЕЙТИНГОВОЙ РАБОТЫ

Для оценки рейтинговой работы критерии следующие:

**От 85 до 100 баллов** ставится, если соблюдены все требования к выполнению рейтинговой работы, сформированы файлы результатов, соблюдены требования к внешнему оформлению рейтинговой работы.

**От 66 до 84 баллов** – основные требования к выполнению рейтинговой работы соблюдены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; имеются упущения в оформлении.

**От 50 до 65 баллов** – имеются существенные отступления от требований к содержанию рейтинговой работы. В частности: допущены логические ошибки в алгоритмах сортировки, выдается неверный результат, производится неверная обработка файла исходных данных, используются встроенные функции для сортировки набора данных

**49 баллов и менее** – обнаруживаются существенные ошибки в программном коде, для сортировки вместо двух использовался один алгоритм, при попытке корректно изменить данные в файле исходных данных – интерпретатор Python выдает ошибки, файл результатов имеет неверный формат или не формируется вообще, отсутствуют подробные комментарии к каждому оператору, отсутствует файл программы \*.py

Основные критерии и показатели оценки расчетно-аналитического задания

| **Критерии** | **Показатели** |
| --- | --- |
| Применение навыков по построению алгоритма решения поставленной задачи. | Верно составленная алгоритмическая модель с использованием встроенных функций Python. В данной рейтинговой работе запрещено использование встроенных функций сортировки (.sort(), функции sorted(), reverse() и т.п.), за исключением самопроверки, на этапе тестирования результатов работы двух алгоритмов сортировки. |
| Владение языком дисциплины (понятийно-категориальным аппаратом) | Четкое и полное определение рассматриваемых понятий (категорий), приводя соответствующие примеры в строгом соответствии с рассматриваемой проблемой |
| Самостоятельный и обоснованный выбор алгоритма сортировки | Выбранный алгоритм сортировки привел к правильному результату, который может быть проверен с использованием стандартных функций языка. |
| Логическая последовательность изложения. | Соблюдение этапов и алгоритма решения задач. |
| Аккуратность оформления и корректность цитирования | Оформление текста с полным соблюдением правил русской орфографии и пунктуации, методических требований и ГОСТа. |

Балльная шкала для оценки рейтинговой работы

| **Позиция**  **реферата** | **Макс. количество баллов**  **(max = 100)** | **Содержание элементов оценки**  **и критерии распределения баллов** |
| --- | --- | --- |
| Наличие условия задачи | 10 | - полное изложение условия -10; |
| Основная часть | 65 | **-** построен алгоритм решения задачи – максимум 20 баллов;  - код программы написан без ошибок – максимум 40 баллов;  - выписан ответ – максимум 5 баллов; |
| Выводы | 10 | **-** приведен полный вывод задач. |
| Список литературы | 5 | - соответствие использованной литературы теме работы – максимум 5 балла; |
| Оформление работы | 5 | Соответствие ГОСТу и требованиям ВУЗа |
| Программный код | 5 | Соответствие требованиям Соглашения PEP 8 |

# 8. ПРИЛОЖЕНИЕ

**Титульный лист рейтинговой работы**



***Кафедра информационных систем***

| ***Рейтинговая работа*** | ***Расчетно-аналитическое задание*** |
| --- | --- |
| ***по дисциплине*** | ***Алгоритмизация и программирование*** |
|  |  |
| ***Задание / Вариант №*** | ***Индивидуальное задание в соответствие с ФИО и ID*** |
|  |  |
| ***Тема*** | ***Алгоритмы типовых задач*** |
|  |  |
| ***Выполнена обучающимся группы*** |  |
| ***ФИО обучающегося*** |  |
|  |  |
| ***Преподаватель*** |  |

Москва – 201\_\_ г.