

# **EXTERNAL JAVA TRAINING. TASK INFORMATION HANDLING**

---

---

[www.training.by](http://www.training.by)

**Legal Notice:** This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM Systems.

---

May 18, 2021

## Task. Information handling

*Создать приложение, разбирающее текст из файла и выполняющее с текстом пять различных операций.*

### Требования

- Разобранный текст должен быть представлен в виде объекта-структуры, содержащего, например, абзацы, предложения, лексемы, слова, выражения, символы. *Лексема - часть текста, ограниченная пробельными символами.* Для организации структуры данных использовать паттерн **Composite**.
- Классы с информацией являются классами сущностей и не должны быть перенагружены методами логики.
- Исходный текст **всегда корректный**. То есть, все предложения начинаются с заглавной буквы и завершаются символами «.», «?», «!» или «...». Все абзацы начинаются с символа табуляции или заданного числа пробелов, например 4 пробела.
- Текст из Composite необходимо восстановить в текстовом виде. Пробелы и знаки табуляции при разборе могут заменяться одним пробелом.
- Для деления текста на составляющие следует использовать регулярные выражения. Не забывать, что регулярные выражения для приложения являются литеральными константами.
- Код, выполняющий разбиение текста на составляющие части, следует оформить в виде классов-парсеров с использованием паттерна **Chain of Responsibility**.
- При разработке парсеров, разбирающих текст, необходимо следить за количеством создаваемых объектов-парсеров. Их не должно быть слишком много.
- (optional) *Битовые выражения, встречающиеся в тексте, должны быть вычислены. И в итоговый текст (структуру данных) должно войти вычисленное значение. Использовать паттерн **Interpreter** с применением функциональных интерфейсов.*
- Для записи логов использовать Log4J2.
- Созданное приложение должно позволять реализовывать функционал по работе над текстом (задачи приведены ниже) без “переписывания” существующего кода.
- Код должен быть покрыт тестами.
- Класс с методом main в задании должен отсутствовать. Запуск только с применением тестов.

### Функциональные возможности, варианты для реализации

- 1 Отсортировать абзацы по количеству предложений.
- 2 Найти предложения с самым длинным словом.
- 3 Удалить из текста все предложения с числом слов меньше заданного.
- 4 Найти в тексте все одинаковые слова без учета регистра и посчитать их количество.

## 5 Подсчитать в предложении число гласных и согласных букв.

- **Пример текста для обработки**

It has survived - not only (five) centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the “Динамо” (Рига) with the release of Letraset sheets.toString() containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker Faclon9 including versions of Lorem Ipsum!

It is a long a!=b established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Ipsum is that it has a more-or-less normal distribution ob.toString(a?b:c), as opposed to using (Content here), content here's, making it look like readable English?

It is a established fact that a reader will be of a page when looking at its layout...

Bye бандерлоги.

- **(optional)Пример текста для обработки**

It has survived - not only (five) centuries, but also the leap into 13<<2 electronic typesetting, remaining 3>>5 essentially ~6&9|(3&4) unchanged. It was popularised in the 5|(1&2&(3|(4&(1^5|6&47)|3)|(~89&4|(42&7)))|1) with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using (~71&(2&3|(3|(2&1>>2|2)&2)|10&2))|78 Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using (Content here), content here', making it look like readable English.

It is a (7^5|1&2<<(2|5>>2&71))|1200 established fact that a reader will be of a page when looking at its layout.

Bye.

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
<1.0>	Первая версия	Игорь Блинов Ольга Смолякова	08-05-2014		
<1.1>	Незначительные правки формулировок предложений	Ольга Смолякова	24-09-2015		
<1.2>	Изменения общей формулировки задачи. Удаление устаревших вариантов. Добавление новых вариантов	Игорь Блинов	29-09-2015		
<1.3>	Добавление нового варианта задания. Уточнение общей формулировки	Игорь Блинов	06-10-2016		
<1.4>	Удаление варианта задания. Уточнение общей формулировки и некоторых заданий	Игорь Блинов	28-11-2016		
<1.5>	Изменение формулировок 4-х заданий	Игорь Блинов	05-02-2018		