**Опис програми**

Ця програма моделює обробку тексту з використанням об'єктно-орієнтованого підходу. Вона розбиває текст на окремі об'єкти, такі як **слова**, **речення**, **букви**, та **знаки пунктуації**, і дозволяє виконувати операції над ними, наприклад, сортування слів за кількістю входжень заданого символу.

**Основні класи та їх можливості**

**1. Клас Letter**

* Модель літери з одним полем value, що зберігає символ.
* Метод getValue() повертає значення літери.

**2. Клас Punctuation**

* Модель знаку пунктуації.
* Поле mark містить символ пунктуації.
* Метод toString() повертає знак як рядок.

**3. Клас Word**

* Модель слова, представленого списком об'єктів Letter.
* Основні методи:
  + **countOccurrences(char targetChar)**: Рахує кількість входжень символу у слові.
  + **toString()**: Повертає слово у вигляді рядка.

**4. Клас Sentence**

* Модель речення, яке складається зі слів та знаків пунктуації.
* Основні методи:
  + **getWords()**: Повертає список всіх слів у реченні.
  + **toString()**: Формує рядкове представлення речення.

**5. Клас Text**

* Модель тексту, що містить список речень.
* Основні методи:
  + **getAllWords()**: Повертає всі слова з усіх речень тексту.
  + **toString()**: Формує рядкове представлення тексту.

**6. Клас TextProcessor**

* Програма для обробки тексту. Використовує класи Text, Sentence, та Word.
* Основний алгоритм:
  1. Розбиває текст на речення та слова.
  2. Сортує слова за кількістю входжень заданого символу.
  3. Виводить результати.

**Приклад роботи**

**Вхідні дані**

String inputText = "REFEREED, CREEPERED, EMEER, DEFERENCE, REFEREE, DEEPFREEZE, BEEKEEPER.";

char targetChar = 'E';

**Вихід**

Sorted words:

EMEER

REFEREE

DEFERENCE

REFEREED

CREEPERED

BEEKEEPER

DEEPFREEZE

**Як це працює**

1. **Парсинг тексту (Text)**:
   * Розбиває текст на речення, використовуючи регулярний вираз (?<=[.!?])\s+.
   * Кожне речення обробляється об'єктом Sentence.
2. **Парсинг речення (Sentence)**:
   * Розбиває речення на слова та пунктуацію, використовуючи регулярний вираз (?=\\p{Punct})|(?<=\\p{Punct})|\\s+.
   * Слова зберігаються як об'єкти Word, а пунктуація — як Punctuation.
3. **Сортування (TextProcessor)**:
   * Всі слова збираються у список через метод getAllWords().
   * Список сортується за кількістю входжень символу E методом countOccurrences(char).

**Ключові аспекти**

* **Об'єктна декомпозиція**:
  + Кожен елемент тексту (буква, слово, речення) має власний клас із чітко визначеною відповідальністю.
* **Масштабованість**:
  + Класи легко розширювати, додаючи нові методи (наприклад, аналіз довжини слова, пошук символів тощо).
* **Регулярні вирази**:
  + Використовуються для ефективного парсингу тексту.
* **Сортування**:
  + Сортування виконується через Comparator для динамічного вибору критерію.