УДК 004.4; 004.6

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА «ГЕНЕРАТОР СЛОВАРЯ»

Т.В. ГУСАКОВ
(Представлено: С.Г. СУРТО)

Введение. В современном мире программный продукт перед выпуском на рынок должен пройти десятки, если не сотни, различных проверок, начиная от Unit тестирования, заканчивая ориентированием пользователя в интерфейсе данного продукта. И главной проблемой зачастую является нехватка данных для проверки функционала и работоспособности данного функционала. К примеру, если проверять работу Back-End'а сайта, нужно иметь «тестовые аккаунты», записанные в базе данных. Если проверять калькулятор – нужны уравнения, которые можно будет в него записать и т.д. С такой проблемой столкнулись и мы при проектировании кластера обработки так называемой Big Data.

Большие данные (англ. big data) – обозначение структурированных и неструктурированных данных огромных объёмов и значительного многообразия, эффективно обрабатываемых горизонтально масштабируемыми программными инструментами [1].

Из-за того, что, по понятным причинам, достать информацию реальных пользователей или опросов крайне дорогое или порой невозможное действие, было решено создать приложение, которое брало бы данные, а конкретно слова, из предоставленного словаря и генерировало новые, случайные данные. При некоторой модернизации и доработки этого приложения можно получать объекты, которые требует именно ваша программа или задача.

Выбор языка программирования. Данное программное обеспечение написано с использованием объектно-ориентированного языка программирования общего назначения – Java. Именно этот язык программирования был выбран нами не случайно, ведь он имеет множество преимуществ среди других языков.

Как выше уже было сказано, Java – объектно-ориентированный язык программирования, что позволяет не только определять тип структуры, но и набор функций, которые мы можем применять к конкретному объекту. Таким образом, структура данных становится объектом, которым можно управлять для создания отношений между различными объектами.

Одним из немаловажных преимуществ Java является Garbage collection. Сборка мусора (англ. garbage collection) в программировании – одна из форм автоматического управления памятью. Специальный процесс, называемый сборщиком мусора (англ. garbage collector), периодически освобождает память, удаляя объекты, которые уже не будут востребованы приложениями [2]. Что позволяет нам не переживать за переполнение памяти и не нагружать код командами отчистки памяти и её выделения.

Ещё одно и самое серьёзное преимущество Java среди других языков – кроссплатформенность. Кроссплатформенность (межплатформенность) – способность программного обеспечения работать с несколькими аппаратными платформами или операционными системами. Обеспечивается благодаря использованию высокоуровневых языков программирования, сред разработки и выполнения, поддерживающих условную компиляцию, компоновку и выполнение кода для различных платформ [3].

О самом программном обеспечении. Главная её задача случайным образом выбирать слова из предоставленного словаря. Словарём может являться книга, стихотворение и т.п. Объём требуемого текста, а также словарь, указывается параметрами запуска приложения.

Основной функцией приложения является считывание всех возможных слов из словаря с указанием минимальной длины слова. Реализация данного процесса использует данные в байтах. Это сделано для универсальности данного программного обеспечения и решения проблем с различными кодировками текста.

Первым делом программа находит в словаре все уникальные слова и заносит их в «базу», откуда в последствии будет их брать. База реализована с использованием ArrayList по той простой причине, что данная коллекция (коллекция – это объект, способный хранить группу одинаковых элементов) имеет быстрый доступ к объекту по его индексу, что в случае случайной выборки позволит нам ускорить выполнение нашей программы. Эта функция представлена в листинге 1.1.

Листинг 1.1. – Функция создания словаря с уникальными словами

```
private List<byte[]> generateList(byte[] array, int minLen) {
List<byte[]> ret = new ArrayList<>();
int i = 0;
while(i < array.length){
   int idx = lookNextToken(array, i);
   if(idx < 0 || ++idx >= array.length){
      break;
```

```
        }else{
            int idxEnd = lookNextToken(array, idx);
            if(idxEnd < 0 || (idxEnd + 1) >= array.length){
                break;
            }
            if((idxEnd - idx) >= minLen){
                ret.add(Arrays.copyOfRange(array, idx, idxEnd));
            }
            i = idxEnd;
        }
    }
    return ret;
}
```

Функция lookNextToken() проверяет, является ли следующий символ не буквой (листинг 1.2).

Листинг 1.2. – Функция lookNextToken().

```
private int lookNextToken(byte[] array, int idx){
    for(int i = idx; i < array.length; i++){
        if(isFromTokenList(array[i])){
            return i;
        }
    }
    return -1;
}
```

Функция isTokenFromList() проверяет является ли переданный в функцию символом из списка ('SPACE', ',', '.', ';', ':', '\'', '\"', '-', '_', 13, 10, '!', '?) (листинг 1.3).

Листинг 1.3. – Функция `isFromTokenList` ()

```
private boolean isFromTokenList(byte b){
    for(int i = 0; i < TOKENS.length; i++){
        if(TOKENS[i] == b){
            return true;
        }
    }
    return false;
}
```

После создания словаря остаётся случайным образом выбирать из него слова и добавлять их в файл/строку/объект для дальнейшего использования другими программами и обработке таких данных.

Заключение. Главная цель разработки данного программного обеспечения – использование его в тестировании системы, работающей с Big Data. Так как в нашем случае для использования и тестирования системы нам требуется лишь файл больших размеров с различными повторяющимися словами, нынешних функций данного программного обеспечения достаточно, для достижения наших целей.

ЛИТЕРАТУРА

1. [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Big_data – Дата доступа: 17.09.2021.
2. [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Garbage_collection_(computer_science) – Дата доступа: 17.09.2021.
3. [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Cross-platform_software – Дата доступа: 17.09.2021.