Макаров Дмитрий Александрович
магистрант
ФГБОУ ВО «Армавирский государственный
педагогический университет»
г. Армавир, Краснодарский край
Коновалова Виктория Александровна
учитель обществознания
МАОУ – СОШ №7 им. Г.К. Жукова
г. Армавир, Краснодарский край

JAVA COLLECTIONS FRAMEWORK – ИНТЕРФЕЙС SET

***Аннотация**: в данной статье рассмотрен интерфейс Set и классы, предоставляющие реализацию данного интерфейса. Большинство классов предоставляют полную реализацию соответствующих интерфейсов и могут использоваться без изменений.*

***Ключевые слова**: каркас коллекций, интерфейс Set, метод add ().*

Каркас коллекций Collections Framework в Java стандартизирует способы управления группами объектов в прикладных программах. Коллекции не были частью исходной версии языка Java, но были внедрены в версии J2SE-1.2. Каркас коллекций был разработан для достижения нескольких целей. Во-первых, он должен был обеспечивать высокую производительность. Во-вторых, каркас должен был обеспечивать единообразное функционирование коллекций с высокой степенью взаимодействия. В-третьих, коллекции должны были допускать простое расширение и/или адаптацию. В каркасе коллекций определяется несколько интерфейсов.

В интерфейсе Set определяется множество. Он расширяет интерфейс Collection и определяет поведение коллекций, не допускающих дублирования элементов. Таким образом, метод add () возвращает логическое значение false при попытке ввести в множество дублирующий элемент. В этом интерфейсе не определяется никаких дополнительных методов.

Так как Set является интерфейсом, вам необходимо создать экземпляр с конкретной реализацией интерфейса для того, чтобы его использовать. Вы можете выбрать из API Java Collections один из следующих вариантов реализаций Set.

– java.util.EnumSet;

– java.util.HashSet;

– java.util.LinkedHashSet;

– java.util.TreeSet.

Ниже представлены примеры как создать экземпляр интерфейса Set:

– Set setA = new EnumSet();

– Set setB = new HashSet();

– Set setC = new LinkedHashSet();

– Set setD = new TreeSet();

Добавить элемент в коллекцию и получить доступ можно воспользовавшись методом add () наследуемый из интерфейса Collection, ниже представлен пример реализации:

```
Set setA = new HashSet();
setA.add(«element 1»);
setA.add(«element 2»);
setA.add(«element 3»);
```

По умолчанию вы можете поместить в набор любой объект, но с Java 5, механизм обобщения позволяет ограничить типы объекта, которые можно вставить в набор. Вот пример:

```
Set<String> setA = new HashSet();
setA.add(«element 1»);
setA.add(«element 2»);
setA.add(«element 3»);
```

Получить элементы коллекции можно с помощью интерфейса Iterator.

Реализация обхода коллекции представлена ниже:

```
Set setA = new HashSet();
setA.add(«element 0»);
```

```
setA.add(«element 1»);
setA.add(«element 2»);
//получить доступ к элементам через Iterator
Iterator iterator = setA.iterator();
while(iterator.hasNext(){
String element = (String) iterator.next();
//получить элементы посредством цикла foreach
for(Object object : setA) {
String element = (String) object;
```

При переборе элементов, порядок их вывода зависит от конкретной реализации интерфейса Set. Разберем особенности классов, реализующих данный интерфейс.

Класс HashSet расширяет класс AbstractSet и реализует интерфейс Set. Он служит для создания коллекции, для хранения элементов которой используется хеш-таблица. В классе HashSet не определяется никаких дополнительных методов, помимо тех, что предоставляют его суперклассы и интерфейсы. Класс HashSet не гарантирует упорядоченности элементов.

Класс TreeSet расширяет класс AbstractSet и реализует интерфейс NavigableSet. Он создает коллекцию, где для хранения элементов применяет древовидная структура. Объекты сохраняются в отсортированном порядке по нарастающей.

В классе LinkedHashSet поддерживается связный список элементов хеш-множества в том порядке, в каком они введены в него. Это позволяет организовать итерацию с вводом элементов в определенном порядке. Аналогично работает итерация в классе EnumSet.

В пакет java.util входит одна из самых эффективных подсистем Java – каркас коллекций Collections Framework. В ней представлены интерфейсы коллекций, а также стандартные классы, которые реализуют их. Большинство классов предоставляют полную реализацию соответствующих интерфейсов и могут использоваться без изменений.

Список литературы

1. Interface Set // https://docs.oracle.com [Электронный ресурс]. – Режим доступа: https://docs.oracle.com/javase/7/docs/api/java/util/Set.html