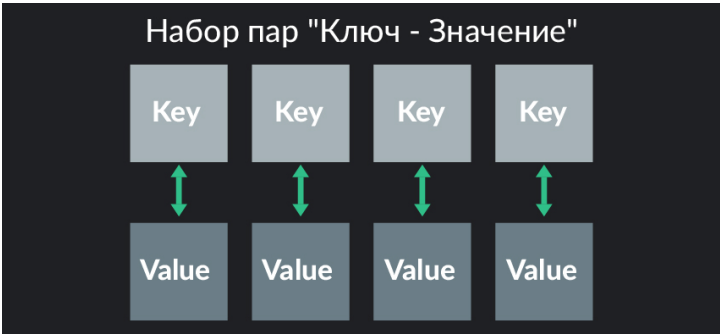




Список всех коллекций

Java Syntax (/quests/QUEST_JAVA_SYNTAX)
8 уровень (/quests/lectures/?quest=QUEST_JAVA_SYNTAX&level=8), 2 лекция (/quests/lectures/questsyntax.level08.lecture02)

ОТКРЫТА



- Привет, Амиго.
- Здорова, Риша.
- Элли мне сказала, что ты хотел больше примеров коллекций. Сейчас я дам тебе несколько. Хочу представить тебе список коллекций и интерфейсов:

Интерфейс

Класс/Реализация

Описание

List

ArrayList

Список

LinkedList

Список

Vector

Вектор

Stack

Стек

Set

HashSet

Множество

TreeSet

Множество

SortedSet

Отсортированное множество

Map

HashMap

Карта/Словарь

TreeMap

Карта/Словарь

SortedMap

Отсортированный слова

Hashtable

Хеш-таблица

— Хм. Как их много. Четыре List'a, три Set'a и четыре Map'a.

— Да, это все различные реализации интерфейсов List, Set и Map.

— А чем отличаются различные реализации?

— Об этом как раз сегодня мы тебе и расскажем. Просто подожди немного.

— Может, у тебя уже накопились какие-нибудь вопросы?

— Как вывести список на экран я знаю. А как вывести Set и Map?

— У элементов списка (List) есть четко заданный порядок, поэтому их можно вывести просто по номеру. У Set и Map строго заданного порядка элементов нет. Собственно говоря, порядок их элементов может меняться при удалении или добавлении какого-нибудь элемента.


— Надо же.

— Поэтому для работы с элементами коллекций были придуманы специальные объекты – итераторы. С их помощью можно пройти по всем элементам коллекции, даже если у них нет номера, а только имена (Map), или вообще нет имён – Set.

-Примеры:

Вывод на экран элементов Set

```
1 public static void main(String[] args)
2 {
3     Set<String> set = new HashSet<String>();
4     set.add("Mama");
5     set.add("Mila");
6     set.add("Ramu");
7
8     //получение итератора для множества
9     Iterator<String> iterator = set.iterator();
10
11     while (iterator.hasNext())        //проверка, есть ли ещё элементы
12     {
13         //получение текущего элемента и переход на следующий
14         String text = iterator.next();
15
16         System.out.println(text);
17     }
18 }
19
```

ЗАДАЧА  Java Syntax, 8 уровень, 2 лекция

ДОСТУПНА

★★★★☆

Набираем код

×10

Иногда думать не надо, строчить надо! Как ни парадоксально звучит, порой пальцы «запоминают» лучше, чем сознание. Вот почему во время обучения в секретном центре JavaRush вы иногда встречаете задания на набор кода. Набирая код, вы привыкаете к синтаксису и зарабатываете немного материи. А ещё — боретесь с ленью.

[Открыть](#)

Вывод на экран элементов List

```
1 public static void main(String[] args)
2 {
3     List<String> list = new ArrayList<String>();
4     list.add("Mama");
5     list.add("Mila");
6     list.add("Ramu");
7
8     Iterator<String> iterator = list.iterator(); //получение итератора для списка
9
10    while (iterator.hasNext())        //проверка, есть ли ещё элементы
11    {
12        //получение текущего элемента и переход на следующий
13        String text = iterator.next();
14
15        System.out.println(text);
16    }
17 }
```

Вывод на экран элементов Map

```
1 public static void main(String[] args)
2 {
3     //все элементы хранятся в парах
4     Map<String, String> map = new HashMap<String, String>();
5     map.put("first", "Mama");
6     map.put("second", "Mila");
7     map.put("third", "Ramu");
8
9     Iterator<Map.Entry<String, String>> iterator = map.entrySet().iterator();
10
11     while (iterator.hasNext())
12     {
13         //получение «пары» элементов
14         Map.Entry<String, String> pair = iterator.next();
15         String key = pair.getKey();           //ключ
16         String value = pair.getValue();       //значение
17         System.out.println(key + ":" + value);
18     }
19 }
```

— Ничего себе! И что все это значит?

— На самом деле все довольно просто. Сначала мы получаем у коллекции специальный объект-iterator. У него есть всего два метода

1. Метод `next()` возвращает очередной элемент коллекции.

2. Метод `hasNext()` проверяет, есть ли еще элементы, которые не возвращал `next()`.

— Ага. Похоже ситуация начинает проявляться. Давай я расскажу, как я все это понял.

— Так. Сначала надо вызвать у коллекции метод `iterator()`, чтобы получить этот волшебный объект-итератор.

— Затем мы в цикле, пока есть еще неполученные элементы, получаем их по одному. Получаем мы очередной элемент коллекции с помощью вызова `next()`, а проверяем, есть ли еще элементы в итераторе с помощью `hasNext()`. Все верно?

— Да, примерно все так и есть. Но самое интересное будет сейчас.

— В Java есть сокращённая запись работы с итераторами. По аналогии с `while`, `for` был добавлен еще один специальный оператор «for each» — «для каждого». Обозначается тоже ключевым словом `for`.

— Оператор `for-each` используется только при работе с коллекциями и контейнерами. В нем неявно используется итератор, но мы видим уже полученный элемент.

— Давай я покажу тебе длинную и сокращенную работу с итератором:

Длинная запись

```
1 public static void main(String[] args)
2 {
3     Set<String> set = new HashSet<String>();
4     set.add("Mama");
5     set.add("Mila");
6     set.add("Ramu");
7
8     Iterator<String> iterator = set.iterator();
9     while (iterator.hasNext())
10    {
11        String text = iterator.next();
12        System.out.println(text);
13    }
14 }
```

Сокращенная запись

```
1 public static void main(String[] args)
2 {
3     Set<String> set = new HashSet<String>();
4     set.add("Mama");
5     set.add("Mila");
6     set.add("Ramu");
7
8     for (String text : set)
9     {
10         System.out.println(text);
11     }
12 }
13
14
```

— Обрати внимание: в правой таблице нет ни зелёных, ни красных слов. Фактически 3 строки заменяются на одну:

Длинная запись

```
1 Iterator<String> iterator = set.iterator();
2 while (iterator.hasNext())
3 {
4     String text = iterator.next();
```

Сокращенная запись

```
1
2 for (String text : set)
3
4
```

— Выглядит шикарно. Так мне нравится гораздо больше!

— Давай посмотрим на те же примеры, что и выше, только записанные короче:

Вывод на экран элементов Set

```
1 public static void main(String[] args)
2 {
3     Set<String> set = new HashSet<String>();
4     set.add("Mama");
5     set.add("Mila");
6     set.add("Ramu");
7
8     for (String text : set)
9     {
10         System.out.println(text);
11     }
12 }
```

Вывод на экран элементов List

```
1 public static void main(String[] args)
2 {
3     List<String> list = new ArrayList<String>();
4     list.add("Mama");
5     list.add("Mila");
6     list.add("Ramu");
7
8     for (String text : list)
9     {
10         System.out.println(text);
11     }
12 }
```

Вывод на экран элементов Map

```
1 public static void main(String[] args)
2 {
3     Map<String, String> map = new HashMap<String, String>();
4     map.put("first", "Mama");
5     map.put("second", "Mila");
6     map.put("third", "Ramu");
7
8     for (Map.Entry<String, String> pair : map.entrySet())
9     {
10         String key = pair.getKey();           //ключ
11         String value = pair.getValue();       //значение
12         System.out.println(key + ":" + value);
13     }
14 }
```

— Так это же другое дело!

— Рад, что тебе понравилось.

[< \(/quests/lectures/questsyntax.level08.lecture01\)](/quests/lectures/questsyntax.level08.lecture01)

[×8 > \(/quests/lectures/questsyntax.level08.lecture03\)](/quests/lectures/questsyntax.level08.lecture03)

 0  0  0  0  0

+46

Комментарии (70)

популярные

новые

старые

Никита

TRIGUN 8 уровень, Москва

9 часов назад

...

Ответить

0

Умеджон Давлатов 8 уровень

2 марта, 00:16

...

Ответить

+6

Витёк (Vitek) 8 уровень

6 февраля, 11:24

...

[Статья на Хабре Java Collections](#)

тут поподробнее, кому интересно
+ все нарисовано :D

Ответить

+12

Абсурд 14 уровень

1 февраля, 02:56

...

Ответить

0

Абсурд 14 уровень

1 февраля, 02:53

...

Такой вопрос. То есть в List можно использовать get(i) а в остальных нельзя? и если мы вводим в set только один элемент то пользоваться для вывода можно только iterator.next() ? Зачем мне переходить к следующему элементу если его нет. Запутанно все

Ответить

0

Витёк (Vitek) 8 уровень

6 февраля, 11:17

...

Интерфейс List

Реализации этого интерфейса представляют собой упорядоченные коллекции. Кроме того, разработчику предоставляется возможность доступа к элементам коллекции по индексу и по значению.

<https://habrahabr.ru/post/237043/>

— У элементов списка (List) есть четко заданный порядок, поэтому их можно вывести просто по номеру. У Set и Map строго заданного порядка элементов нет.

Следовательно, при использовании интерфейсов Set и Map использование get(i) не предусмотрено.

(Рассказал, как понял. Поправьте, если где накосячил :)

Ответить

+4

Nyuruko 18 уровень, Москва

12 января, 20:00

...

Очень много ошибок. Скопируйте в вашу IDE и убедитесь сами. Тут они несколько раз не указали тип данных, с которыми работают:

```
1 Set set = new HashSet();
```

Нужно добавить <String>, и тоже самое для

```
1 Iterator iterator = set.iterator();
```

Ответить

+4

Дмитрий Стуков 17 уровень, Санкт-Петербург

19 января, 12:19

...

поддерживаю, моментами написано реально поверхностно, это напрягает новичка и отбивает желание порой, но надо преодолевать трудности

Ответить

+4

Ответить

+3

Денис 22 уровень, Москва

8 декабря 2017, 17:25

Видимо я где-то проморгал. Почему написано List list = new ArrayList();
А не ArrayList list = new ArrayList();
??

Ответить

+2

Chanting Wolf 14 уровень, Киев

16 декабря 2017, 11:52

List это интерфейс
Хорошая практика использовать переменные типа интерфейса а не конкретного класса
Допустим вы по каким либо причинам захотите поменять скажем ArrayList на LinkedList
Тогда в случае использования интерфейсов надо будет произвести изменение лишь в одном месте и логика останется без изменений

Ответить

+10

Кирилл Рязжский 15 уровень, Москва

10 ноября 2017, 21:30

получаем у коллекции специальный объект-iterator. У него есть всего два метода Ну это ппц. Там не два метода. Это для итераций вперед по списку всего 2 надо. А так их там куда больше.

Ответить

+2

Thinker 20 уровень

29 октября 2017, 19:23

— Обрати внимание: в правой таблице нет ни зелёных, ни красных слов. Фактически 3 строки заменяются на одну === а где есть такие цветные слова в левой таблице? Или я перестал различать цвета?)))

Ответить

+8

Кирилл Рязжский 15 уровень, Москва

10 ноября 2017, 21:35

Раньше там была картинка. Потом они все перекроили, сайт теперь совсем другой, а вот текст свести с кодом видеть забыли. Это еще фигня. У них есть лютые синтаксические ошибки, даже администрации писал, но они не исправляют. Мне эта больше всего нравится

```
1 public static void String name;
```

Ответить

+12

Thinker 20 уровень

18 ноября 2017, 01:16

это пять!))
видимо, джавараш очень экономит на тестировщиках...
совсем без знания азов программирования

Ответить

0

Кирилл Рязжский 15 уровень, Москва

19 ноября 2017, 10:49

Даже нашел, где это писал, вот В 4 лекции 2 уровня в коде примера грубая ошибка, private static void String TEXT = "The end."

Ответить

+1

Vitalis11 8 уровень, Киев

10 октября 2017, 16:14

Последняя бесплатная лекция :-(

Ответить

+3

 Загрузить еще

[jsh.ru/](https://javarush.ru/), [G+ \(https://plus.google.com/114772402300089087607/\)](https://plus.google.com/114772402300089087607/), [_ \(https://twitter.com/javarush_ru/\)](https://twitter.com/javarush_ru/), [Facebook](https://www.facebook.com/javarush.ru/)



Программистами не рождаются
© 2018