火龙果•整理
uml.org.cn

JAVA编程高级
——集合类

Programming Your Future

火龙果•整理
uml.org.cn

本章目标

- 掌握集合框架包含的内容
- 掌握ArrayList和LinkedList的使用
- 掌握HashMap的使用
- 掌握Iterator的使用
- 掌握泛型集合的使用

火龙果•整理
uml.org.cn

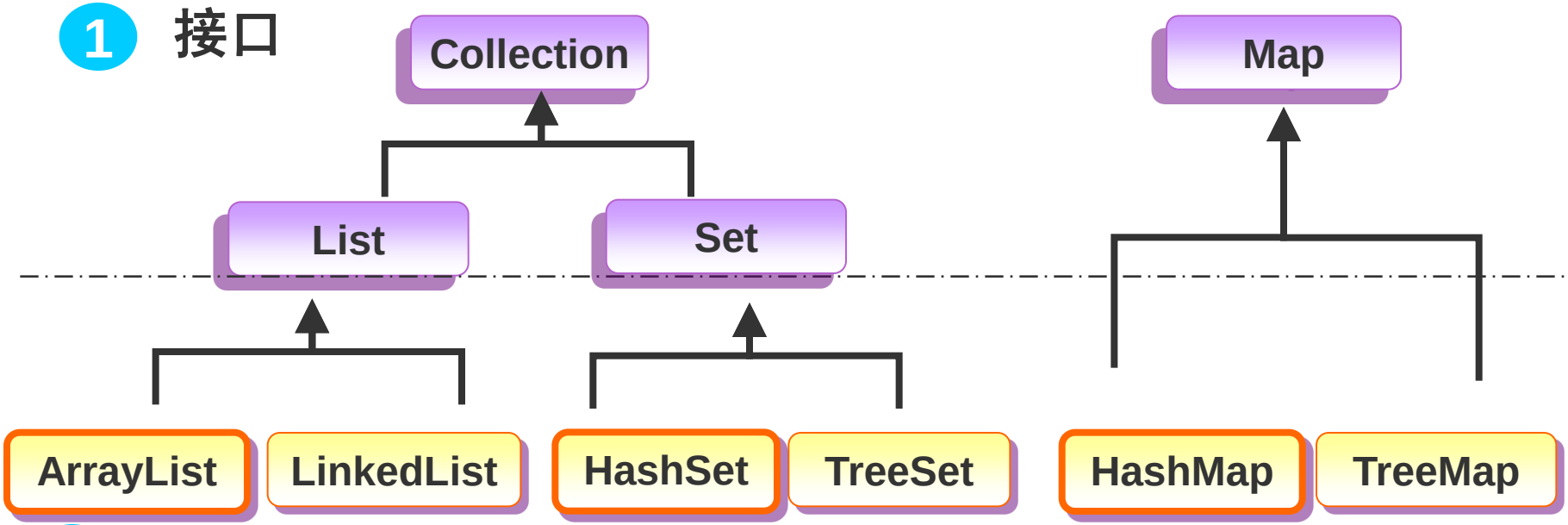
Java中的集合类

- **集合的概念**
  - ✓ Java中集合类是用来存放对象的
  - ✓ 集合相当于一个容器，里面包容着一组对象 —— 容器类
  - ✓ 其中的每个对象作为集合的一个元素出现
  - ✓ Java API提供的集合类位于java.util包内
- **Java中数组与集合的比较**
  - ✓ 数组也是容器，它是定长的，访问较快，但是数组不会自动扩充
  - ✓ 数组可以包含基本数据类型或引用类型的对象，而集合中只能包含引用类型的对象

火龙果•整理
uml.org.cn

Java集合框架包含的内容2-1

1 接口



2 具体类

提供了对集合进行排序、遍历等多种算法实现

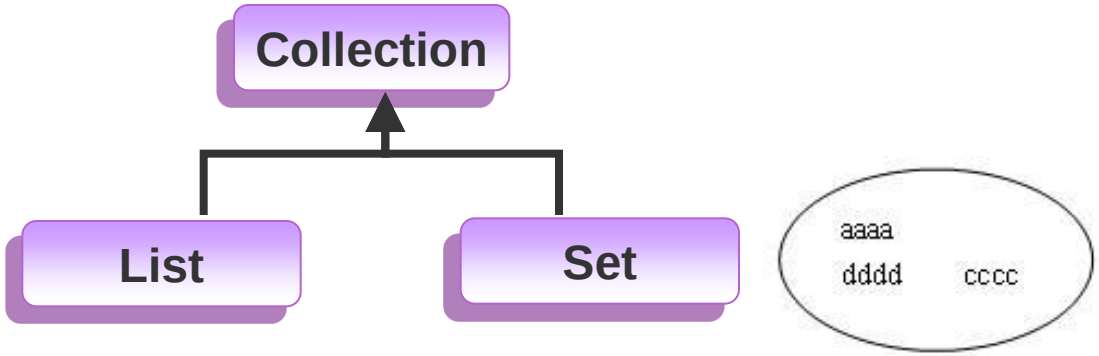
3 算法

Collections

Java集合框架为我们提供了一套性能优良、使用方便的接口和类，它们位于java.util包中

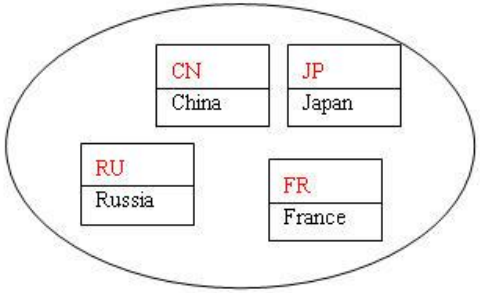
我们不必再重新发明轮子，只需学会如何使用它们，就可处理实际应用中问题

Java集合框架包含的内容2-2



0	1	2	3	4	5	
aaaa	dddd	cccc	aaaa	eeee	dddd	

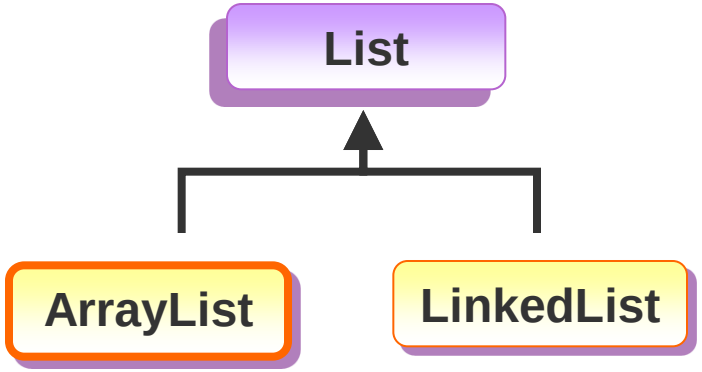
- Collection 接口存储一组不唯一，无序的对象
- List 接口存储一组不唯一，有序（插入顺序）的对象
- Set 接口存储一组唯一，无序的对象



- Map接口存储一组键值对象，提供key到value的映射

火龙果•整理
uml.org.cn

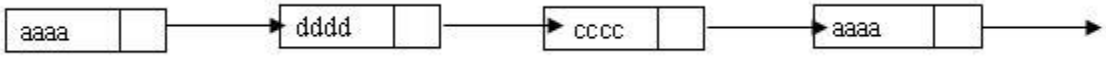
List接口的实现类



- ArrayList实现了长度可变的数组，在内存中分配连续的空间。遍历元素和随机访问元素的效率比较高

0	1	2	3	4	5	
aaaa	dddd	cccc	aaaa	eeee	dddd	

- LinkedList采用链表存储方式。插入、删除元素时效率比较高

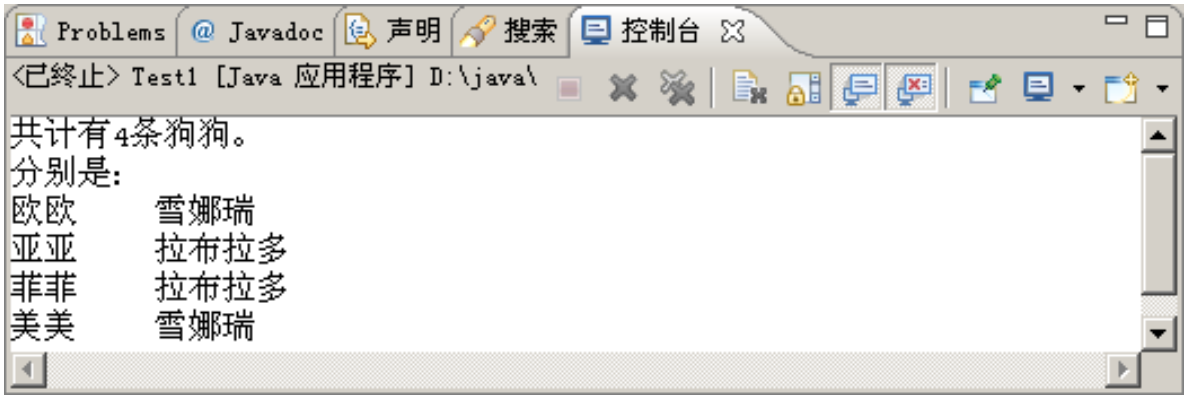


火龙果•整理
uml.org.cn

ArrayList集合类5-1

问题

- 存储多条狗狗信息，获取狗狗总数，逐条打印出各条狗狗信息



分析

- 通过List接口的实现类ArrayList实现该需求
  - 元素个数不确定
  - 要求获得元素的实际个数
  - 按照存储顺序获取并打印元素信息

火龙果•整理
uml.org.cn

ArrayList集合类5-2

示例

```
public class Test1 {
    public static void main(String[] args) {
        Dog ououDog = new Dog("欧欧", "雪娜瑞");
        Dog yayaDog = new Dog
        Dog meimeiDog = new D
        Dog feifeiDog = new Dog

        List dogs = new ArrayList();
        dogs.add(ououDog);
        dogs.add(yayaDog);
        dogs.add(meimeiDog);
        dogs.add(2, feifeiDog); // 添加feifei

        System.out.println("共计有" + dogs.size() + "条狗狗。");

        System.out.println("分别是：");
        for (int i = 0; i < dogs.size(); i++) {
            Dog dog = (Dog) dogs.get(i);
            System.out.println(dog.getName() + "\t"+ dog.getStrain());
        }
    }
}
```

创建ArrayList对象并把四个狗狗对象放入其中

输出狗狗的数量

获取第i个元素

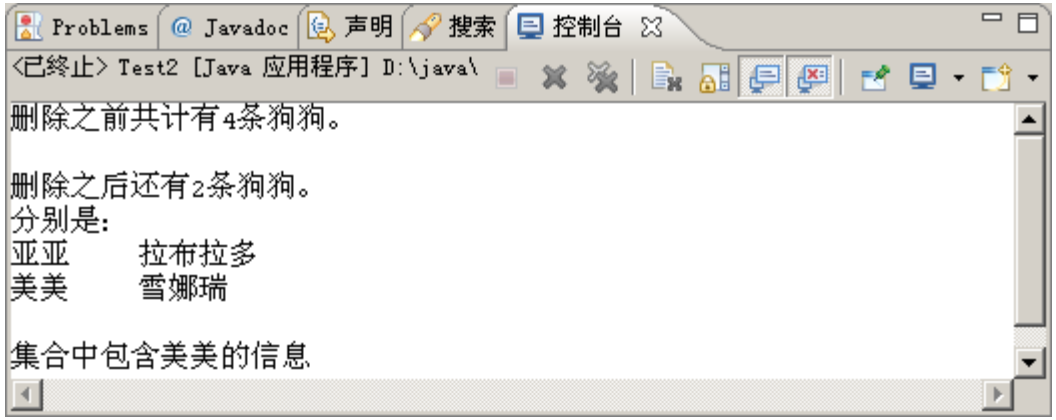
演示示例：使用ArrayList集合类

火龙果•整理
uml.org.cn

ArrayList集合类5-3

问题

- 扩充以下几部分功能
  - 删除指定位置的狗狗，如第一个狗狗
  - 删除指定的狗狗，如删除feifeiDog对象
  - 判断集合中是否包含指定狗狗



- List接口提供相应方法remove()、contains()，直接使用即可

分析

火龙果•整理
uml.org.cn

ArrayList集合类5-4

示例

```
public class Test2 {
    public static void main(String[] args) {
        // 1、创建多个狗狗对象
        …
        // 2、创建ArrayList集合对象并把多个狗狗对象放入其中
        …

        System.out.println("删除之前共计有" +dogs.size() + "条狗狗。");

        dogs.remove(0);
        dogs.remove(feifeiDog);

        System.out.println("\n删除之后还有" + dogs.size() + "条狗狗。");
        …

        if(dogs.contains(meimeiDog))
            System.out.println("\n集合中包含美美的信息");
        else
            System.out.println("\n集合中不包含美美的信息");
    }
}
```

删除第一个狗狗和feifeiDog

判断是否包含指定狗狗

演示示例：使用ArrayList集合类

火龙果•整理
uml.org.cn

ArrayList集合类5-5

- List接口常用方法

方法名	说　　明
boolean add(Object o)	在列表的末尾顺序添加元素，起始索引位置从0开始
void add(int index,Object o)	在指定的索引位置添加元素。索引位置必须介于0和列表中元素个数之间
int size()	返回列表中的元素个数
Object get(int index)	返回指定索引位置处的元素。取出的元素是Object类型，使用前需要进行强制类型转换
boolean contains(Object o)	判断列表中是否存在指定元素
boolean remove(Object o)	从列表中删除元素
Object　　remove(int index)	从列表中删除指定位置元素，起始索引位置从0开始

火龙果•整理
uml.org.cn

练习——添加多个企鹅信息到List中

- 需求说明：
  - 把多个企鹅的信息添加到集合中
  - 查看企鹅的数量
  - 遍历所有企鹅的信息
  - 删除集合中部分企鹅的元素
  - 判断集合中是否包含指定企鹅

完成时间：20分钟

火龙果•整理
uml.org.cn

LinkedList集合类3-1

问题

- 在集合任何位置（头部、中间、尾部）添加、获取、删除狗狗对象

```
Problems  @ Javadoc  声明  搜索  控制台
<已终止> Test3 [Java 应用程序] D:\java\
第一条狗狗的昵称是菲菲。
最后一条狗狗的昵称是美美。

删除部分狗狗后还有2条狗狗。
分别是:
欧欧        雪娜瑞
亚亚        拉布拉多
```

分析

- 插入、删除操作频繁时，可使用LinkedList来提高效率
- LinkedList还额外提供对头部和尾部元素进行添加和删除操作的方法

火龙果•整理
uml.org.cn

LinkedList集合类3-2

示例

```
public class Test3 {
    public static void main(String[] args) {
        … // 创建多个狗狗对象

        LinkedList dogs = new LinkedList();
        dogs.add(ououDog);
        dogs.add(yayaDog);
        dogs.addLast(meimeiDog);
        dogs.addFirst(feifeiDog);

        Dog dogFirst= (Dog)dogs.getFirst();
        System.out.println("第一条狗狗昵称是"+dogFirst.getName());

        Dog dogLast= (Dog)dogs.getLast();
        System.out.println("最后一条狗狗昵称是"+dogLast.getName());

        dogs.removeFirst();
        dogs.removeLast();
        …
    }
}
```

创建LinkedList集合对象并把多个狗狗对象放入其中

获取第一条狗狗信息

获取最后一条狗狗信息

删除第一个狗狗和最后一个狗狗

演示示例：使用LinkedList集合类

火龙果•整理
uml.org.cn

LinkedList集合类3-3

- LinkedList的特殊方法

方法名	说　　明
void　addFirst(Object o)	在列表的首部添加元素
void　addLast(Object o)	在列表的末尾添加元素
Object　getFirst()	返回列表中的第一个元素
Object　getLast()	返回列表中的最后一个元素
Object　removeFirst()	删除并返回列表中的第一个元素
Object　removeLast()	删除并返回列表中的最后一个元素

火龙果•整理
uml.org.cn

小结

提问

- 集合框架有何好处？
- Java集合框架中包含哪些接口和哪些类？
- ArrayList和LinkedList有何异同？
- 用纸笔绘制List接口和两个实现类的类图

Map接口3-1

问题

- 建立国家英文简称和中文全名间的键值映射，并通过key对value进行操作，应该如何实现数据的存储和操作呢？

```
Problems  @ Javadoc  声明  搜索  控制台
<已终止> Test4 [Java 应用程序] D:\java\
CN对应的国家是：中华人民共和国
Map中共有4组数据
Map中包含FR的key吗？true
Map中包含FR的key吗？false
[US, RU, CN]
[美利坚合众国, 俄罗斯联邦, 中华人民共和国]
{US=美利坚合众国, RU=俄罗斯联邦, CN=中华人民共和国}
```

分析

- Map接口专门处理键值映射数据的存储，可以根据键实现对值的操作
- 最常用的实现类是HashMap

火龙果•整理
uml.org.cn

Map接口3-2

示例

```
public class Test4 {
    public static void main(String[] args) {

        Map countries = new HashMap();
        countries.put("CN", "中华人民共和国");
        countries.put("RU", "俄罗斯联邦");
        countries.put("FR", "法兰西共和国");
        countries.put("US", "美利坚合众国");

        String country = (String) countries.get("
        System.out.println("CN对应的国家是：" + co);

        System.out.println("Map中共有"+countries.size()+"组数据");

        countries.remove("FR");
        System.out.println("Map中包含FR的key
                countries.containsKey("FR"));

        System.out.println( countries.keySet() ) ;
        System.out.println( countries.values() );
        System.out.println( countries );
    }
}
```

使用HashMap存储多组键值对

获取指定元素的值

获取Map元素个数

删除指定元素，判断是否包含指定元素

显示键集、值集和键值对集

演示示例：使用HashMap集合类

火龙果•整理
uml.org.cn

Map接口3-3

- Map接口常用方法

方法名	说　　明
Object put(Object key, Object val)	以“键-值对”的方式进行存储
Object get (Object key)	根据键返回相关联的值，如果不存在指定的键，返回null
Object remove (Object key)	删除由指定的键映射的“键-值对”
int size()	返回元素个数
Set keySet ()	返回键的集合
Collection values ()	返回值的集合
boolean　containsKey (Object key)	如果存在由指定的键映射的“键-值对”，返回true

火龙果•整理
uml.org.cn

练习——根据宠物昵称查找宠物

- 需求说明：
  - 根据宠物昵称查找对应宠物，如果找到，显示宠物信息，否则给出错误提示

完成时间：15分钟

迭代器Iterator2-1

问题

- 如何遍历List集合呢？

分析

- 方法1：通过for循环和get()方法配合实现遍历
- 方法2：通过迭代器Iterator实现遍历
  - 所有集合接口和类都没有提供相应遍历方法，而是由Iterator实现集合遍历
  - Collection 接口的iterate()方法返回一个Iterator，然后通过Iterator接口的两个方法可实现遍历
    - boolean hasNext(): 判断是否存在另一个可访问的元素
    - Object next(): 返回要访问的下一个元素

火龙果•整理
uml.org.cn

迭代器Iterator2-2

示例

```
public class Test5 {
    public static void main(String[] args) {
        // 1、创建多个狗狗对象
        Dog ououDog = new Dog("欧欧", "雪娜瑞");
        …
        // 2、创建ArrayList集合对象并把多个狗狗对象放入其中
        List dogs = new ArrayList ();
        dogs.add(ououDog);
        …

        System.out.println("使用Iterator遍历，所有狗狗昵称和品种是:");
        Iterator it = dogs.iterator();
        while (it.hasNext()) {
            Dog dog = (Dog) it.next();
            System.out.println(dog.getName()+"\t"+dog.getStrain());
        }
    }
}
```

通过迭代器依次输出所有狗狗的信息

Problems | @ Javadoc | 声明 | 搜索 | 控制台

<已终止> Test5(1) [Java 应用程序] D:\

```
使用Iterator遍历，所有狗狗的昵称和品种分别是:
欧欧    雪娜瑞
亚亚    拉布拉多
菲菲    拉布拉多
美美    雪娜瑞
```

演示示例：使用Iterator遍历集合

火龙果•整理
uml.org.cn

练习——迭代List中企鹅信息

- 需求说明：
  - 使用ArrayList和LinkedList存储多个企鹅信息，然后统一使用Iterator进行遍历

完成时间：15分钟

火龙果•整理
uml.org.cn

泛型集合3-1

问题

- 把任何类型对象通过add(Object obj) 放入List中，认为只是Object类型
- 通过get(int index) 取出List中元素时必须进行强制类型转换，繁琐而且容易出现异常
- 使用Map的put(Object key, Object value)和get (Object key)存取对象时存在同样问题
- 使用Iterator的next()方法获取元素时存在同样问题

分析

- JDK5.0中通过引入泛型有效的解决了这个问题
- JDK5.0使用泛型改写了集合框架中的所有接口和类

火龙果•整理
uml.org.cn

泛型集合3-2

示例

```
public class Test6 {
    public static void main(String[] args) {
        // 1、创建多个狗狗对象
        Dog dog
        欧", "雪娜瑞
        …
        // 2、创建Arr
        把多个狗
        List <Dog> d
        List<Dog>();
        dogs.add(do
        …
        //dogs.add("hello");
        // 3、 显示第三个元素的信
        Dog dog = dogs.get(2);
        System.out.println("第三个狗狗的信息如下：");
        System.out.p
        me() + "\t" + dog.getStrain());
        //4、使用Iterat
        System.out.pri
        ");
        Iterator <Dog> it = d
        while (it.hasNext()) {
            dog = it.next();
            System.out.println(dog.getName() + "\t" + dog.getStrain());
        }
    }
}
```

标记元素类型

标记元素类型

类型不符，出现编译错误

无需类型强制转换
(Dog) Dogs.get(2);

标记元素类型

无需类型强制转换
(Dog) it.next();

演示示例：使用泛型集合

泛型集合3-3

示例

```
public class Test7 {
    public static void ma[illegible] {

        //1、使用HashMap存储多组国家英文简称和中文全称的键值对
        Map <String,String> countries=new HashMap<String,String>();
        countries.put("CN", "中华人民共和国");
        countries.put("RU", "俄罗斯联邦");
        countries.put("FR", "法兰西共和国");

        //2、显示"CN"对应国家的中文全称
        String country= countries.get("CN");
        System.out.println("CN对应的国[illegible]是："+country);
    }
}
```

标记键-值类型

标记键-值类型

无需类型强制转换
(String) countries.get("CN");

演示示例：使用泛型集合

火龙果•整理
uml.org.cn

总结

提问

- Collection 、List 、Set 、Map接口的联系和区别有哪些？
- ArrayList和LinkedList有什么异同之处？
- 有哪些遍历集合的方法？
- 为什么要引入泛型集合？

火龙果•整理
uml.org.cn

小结1

- 创建一个类Cat
  - 包含属性name，在构造方法中进行初始化
  - 添加一个方法show()，用以打印name属性的值
- 创建一个类CatTest，添加main方法，实现
  - 创建一个ArrayList，向其中添加几个Cat对象
  - 遍历该集合，并且对每个Cat对象调用show()方法

火龙果•整理
uml.org.cn

小结2

- 创建一个类Stack，代表堆栈（其特点为：后进先出），添加方法add(Object obj)、以及get()，添加main方法进行验证，要求：
  - 使用LinkedList实现堆栈
  - 在向LinkedList中添加时，使用addLast方法
  - 在从LinkedList中取出时，使用removeLast方法

火龙果•整理
uml.org.cn

小结3

- 创建一个类Book
  - 包含属性：title（标题），使用构造方法进行初始化
  - 重写toString()方法，用以返回Title属性的值
- 创建一个类BookTest，添加main方法，要求：
  - 使用HashMap进行存储，键为Book对象的编号，值为Book对象
  - 通过某一个编号获取Book对象，并打印该Book对象的标题