

day07_Scanner类、Random类、ArrayList类

一.API

API(Application Programming Interface), 应用程序编程接口。Java API是一本程序员的字典, 是JDK中提供给我们使用的类的说明文档。这些类将底层的代码实现封装了起来, 我们不需要关心这些类是如何实现的, 只需要学习这些类如何使用即可。所以我们可以通过查询API文档的方式, 来学习Java提供的类, 并得知如何使用它们。

根据API使用类的一般步骤如下:

(1).导包: 使用import语句导包, 引入要使用的类。

```
import 包路径 . 类名称;
```

- 对于自定义的类: 如果目标类和当前类位于同一个package下, 则此目标类的导包语句可以省略不写。
- 对于JDK中提供的类: 使用API查询类的包路径。只有java.lang包下的类不需要导包语句, 其他包路径的类都需要导包语句。

(2).创建对象: 使用该类的构造方法, 创建一个该类的对象。

(3).调用属性、方法: 使用对象调用该类的成员变量、成员方法, 完成指定的功能。

二.Scanner类

1.Scanner类的使用

Scanner(扫描器), 是JDK当中提供的一个类, 位于java.util包路径下, 是一个可以解析基本类型数据、字符串数据的简单文本扫描器。现阶段我们使用Scanner类来实现, 从键盘输入数据到程序当中。

(1).构造方法:

```
Scanner sc = new Scanner(System.in);
```

 构造方法的参数 System.in 代表从键盘进行输入

(2).成员方法:

```
public int nextInt(): 获取键盘输入的int数字并返回
```

```
public String next(): 获取键盘输入的字符串并返回
```

2.匿名对象 (了解内容)

创建对象时, 只有创建对象的语句, 却没有把对象地址值赋值给某个引用类型变量, 属于创建对象的简化写法。即匿名对象创建时, 只是在堆内存中new一个对象出来, 并没有将其堆内存地址返回给某个引用类型变量存储。匿名对象的创建格式如下:

```
new 类名称(参数列表);
```

(1).匿名对象的使用

匿名对象的使用和普通对象没有什么差别, 匿名对象也可以直接访问成员变量和成员方法。但要注意一点: 每次使用匿名对象的时候, 由于new关键字的存在, 都会创建一个新的对象, 这就导致了匿名对象只能使用一次。

(2).匿名对象作为方法的参数和返回值

三.Random类

Random(随机), 是JDK当中提供的一个类, 位于java.util包路径下, 是一个随机数生成器, 用来生成随机数字。

(1).构造方法:

```
Random r = new Random();
```

 构造方法无参数

(2).成员方法:

```
public int nextInt(): 生成并返回一个随机的int数字, 范围是int所有范围
```

```
public String nextInt(int n): 生成并返回一个随机的int数字, 范围是左闭右开区间 [0,n)
```

四.ArrayList<E>类

1.对象数组

创建一个数组的格式如下：

```
数据类型[] 数组名 = new 数据类型[数组长度];
```

其中的数据类型可以是基本数据类型int、float、double等，也可以是引用数据类型String、类等。当 "Student类" 作为数组的数据类型时，这个 "Student类型数组" 可以用来存储 "Student类" 这一数据类型的数据，即由Student类实例化出来的对象。这种用来存储多个对象的数组，称为对象数组。定义一个对象数组的格式如下：

```
类名[] 数组名 = new 类名[数组长度];
Student[] array = new Student[3];
```

到目前为止，我们想存储对象数据，选择的容器，只有对象数组。而数组的长度是固定的，无法适应数据个数变化的需求。为了解决这个问题，Java提供了另一个存储数据的容器java.util.ArrayList<E>集合类，让我们可以更便捷的存储和操作对象数据。

2.ArrayList<E>集合类的使用

java.util.ArrayList<E>集合类是一个大小可变的，用来存储数据的容器，存储在集合内的数据称为元素。此类提供一些方法来操作存储在集合内的元素。与数组不同的是，ArrayList集合中可不断添加元素，其长度尺寸也自动增长。具体使用如下：

(1).使用ArrayList集合存储字符串

该集合类的全称是java.util.ArrayList<E>，其中<E>表示一种指定的数据类型，叫做泛型。也就是装在集合中的所有元素，全都是统一的什么类型。泛型只能是引用数据类型，不能是基本数据类型。

```
// 创建一个ArrayList<E>集合类的对象，名为list，list对象是一个用来存储字符串类型数据的集合
ArrayList<String> list = new ArrayList<String>();
```

ArrayList<E>集合类中的常用方法：

- public boolean add(E e)：向集合中添加元素，参数的类型和泛型一致，返回值代表是否添加成功
- public E get(int index)：从集合中获取指定索引处的元素，返回值就是对应索引位置的元素（ArrayList<E>集合的索引编号从0开始）
- public E remove(int index)：从集合中删除指定索引处的元素，返回值就是被删除的元素
- public int size()：获取集合的长度，返回值是集合中的元素个数。
- ArrayList<E>集合的遍历操作：调用size()、get()方法

(2).使用ArrayList集合存储对象

当泛型是引用数据类型---"类"的时候，此时ArrayList<E>集合中存储的是"类"这一数据类型的数据，即由类实例化出来的对象。

(3).使用ArrayList集合存储基本数据类型

由于泛型不能是基本数据类型，只能是引用数据类型。所以ArrayList<E>集合不能存储基本类型数据，只能存储引用类型的数据。要想使用ArrayList<E>集合存储基本类型数据，必须要使用基本数据类型对应的包装类。为了满足Java语言面向对象的这一特性，每一个基本类型在java.lang包中都有一个对应的包装类，即将每个基本类型都包装成了一个类。使用ArrayList<包装类>集合来存储基本数据类型的方法如下：

a.基本数据类型对应的包装类：

基本类型(关键字)	包装类
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

b.从JDK1.5+开始，"基本类型"与"包装类型"之间可以发生自动类型转换，称为自动装箱、自动拆箱。

- 自动装箱：基本类型 ---> 包装类型
10是默认的int类型，add(E e)方法中的形参是包装类型。所以调用add方法传递参数时有：E e = 10；此时发生了自动类型转换，10由int类型自动转换为了 "Integer类" 类型，称为自动装箱。
- 自动拆箱：包装类型 ---> 基本类型
get方法的返回值是包装类型，所以调用get方法时有：int num = e；则此行代码发生了自动类型转换，e由 "Integer类" 类型自动转换为了int类型，称为自动拆箱。

3.ArrayList<E>集合类的练习