# **接口静态方法**

interface NewInterface {

// 静态方法

static void hello()

{

System.out.println("Hello, New Static Method Here");

}

// 抽象方法

void overrideMethod(String str);

}

# **接口默认方法**

方法名前面加个 default 关键字即可实现默认方法。

不强迫所有实现类实现该方法，便于接口的扩展。

public interface Vehicle {

default void print(){

System.out.println("我是一辆车!");

}

}

# **Lambda表达式**

userList.stream().forEach(s->System.out.println(s));

# **方法引用**

userList.stream().forEach(System.out::println);

**\*\*Lambda表达式和方法引用应该理解为一个函数而不是一个函数的调用**

# **Stream流**

# **Optional 类**

Optional.of(T t)：创建一个Optional实例

Optional.empty()：创建一个空的Optional实例

Optional.ofNullable(T t)：若t不为null,创建Optional实列，否则创建空实例

isPresent()：判断是否包含值

orElse(T t)：如果调用对象包含值，返回该值，否则返回t

orElseGet(Supplier s)：如果调用对象包含值，返回该值，否则返回s获取的值

map(Function f)：如果有值对其处理，并返回处理后的Optional，否则返回Optional.empty()

flatMap(Function mapper)：与map类似，要求返回值必须是Optional