**学习总结**

姓名 董金伟 专业 计算机科学与技术 班级 2班 日期 2022/1/26

1. 当天所学习主要内容（做一总结）
2. 大整数对象类

计算100以内的阶乘。

不可变的任意精度整数。

BigInteger(String val)

add(BigInteger val)

subtract(BigInteger val)

multiply(BigInteger val)

divide(BigInteger val)

toString()

1. 随机类Random

Random()

nextDouble()：[0.0, 1.0)

nextInt(int bound)：[0, bound)

setSeed(long seed)：设置随机种子

1. 泛型
2. 概念、好处

将类型作为参数进行传递。

类---坐标---整数、浮点数、字符串----Object类型

存在问题：必须强制；编译时不报错；

泛型类 泛型接口 泛型方法

1. 泛型类

方法名(类型名 标识符1，类型名 标识符2，类型名 标识符3……)

class类名<标识符1，标识符2 extends 具体类型，标识符3……>{

数据成员； //E a;

方法成员；

}

定义变量：

类名：Object

类名<实参1，实参2，实参3……>：实参是类型名。不能是基本类型。

实例化对象：

new 类名();

new 类名<实参1，实参2，实参3……>();

注意：

泛型类继承了Object

不能实例化 标识符 对象。如：不能new E();

静态数据成员不能是E类型。

E的对象可以使用instanceof判断是什么类型。如：E a； a instanceof String

可以被继承、重写、多态

方法的参数是泛型类：

class<E> A{}

void f(A<> a){}

A<?>：传递的A对象，A对象的实参是任意类型。

A<? extends 具体类型>：传递的A对象，A对象的实参是具体类型或具体类

型的子类。

A<? super 具体类型>：传递的A对象，A对象的实参是具体类型或具体类

型的父类。

A<具体类型>：传递的A对象，A对象的实参是具体类型。

继承泛型类：

class 子类名<标识符，标识符……> extends 父类名<标识符，标识符……>{

}

class 子类名extends 父类名<具体类型名，具体类型名……>{

}

多态：

和前面讲的多态的使用相同。

抽象泛型类：

泛型类前加abstract

1. 泛型接口

定义泛型接口：

Interface 接口名<标识符，标识符，标识符……>{

数据成员；

方法成员；

}

注意：

数据成员类型不能是标识符。

1. 泛型方法

定义：

访问权限 修饰符 <标识符…> 返回类型 方法名(类型名 参数名，……){

方法体；

}

调用：

对象.<类型名…>方法名(实参表)；

泛型方法可以在 泛型类中 和 非泛型类中 定义。

集合

1. 集合概念、优点：

系统定义的泛型类。通过一定的数据结构保存多个相同类型的对象。

优点：

长度可变

一个元素可以保存两个数据，key value

1. 集合类体系：

Collection：一个元素保存一个数据。

Map：一个元素保存两个数据。key value 键值对

1. 学习中所存在的问题及解决办法

3．学习收获及建议

学习了常用的大整数类，和随机数类，这两个类中的方法只需要记住几个常用的就行。

泛型就是将类型作为参数进行传递。一般都用<>，一对尖括号进行标识。泛型类对象，可以引用不同的类型，但是不能是基本数据类型。

泛型中的抽象和接口和，都与类中一样。