### Java第一阶段—DAY13-JAVA案例

1. 创建Person类，有属性name跟age，重写toString方法、equals方法，并重写clone方法让它具有克隆的特性。
2. public class CommonClassObject {  
     
    public static void main(String[] args) throws Exception {  
    Person p1 = new Person("张三",12);  
    Person p2 = new Person("李四",15);  
    System.out.println(p1 == p2);  
    System.out.println(p1.equals(p2));  
    System.out.println(p1.toString());  
     
    Person p3 = (Person) p1.clone();  
    System.out.println(p1.equals(p3.toString()));  
    }  
     
     
   }  
     
   class Person implements Cloneable {  
     
    private String name;  
    private int age;  
     
    public Person(String name, int age) {  
    this.name = name;  
    this.age = age;  
    }  
     
    @Override  
    public boolean equals(Object o) {  
   *// 如果要比较的2个对象是同一个对象，则直接返回true* if (this == o) return true;  
   *// 如果要比较的2个对象的类型不一致，则直接返回false* if (!(o instanceof Person)) return false;  
   *// 将要比较的对象转成当前类型，并比较姓名是否相同* Person person = (Person) o;  
    return name.equals(person.name);  
    }  
     
    @Override  
    public String toString() {  
    return "Person{" + "name='" + name + '\'' + ", age=" + age + '}';  
    }  
     
    *// 访问权限提升为public，逻辑使用父类的逻辑即可。* @Override  
    public Object clone() throws CloneNotSupportedException {  
    return super.clone();  
    }  
     
   }
3. 练习Math类中的常用方法？

public class CommonClassMath {  
  
 public static void main(String[] args) {  
 System.out.println(Math.PI);*//3.141592653589793* System.out.println(Math.abs(-3.5));*//3.5* System.out.println(Math.ceil(-3.5));*//-3.0* System.out.println(Math.floor(3.5));*//3.0* System.out.println(Math.round(3.5));*//4* System.out.println(Math.round(-3.5));*//-3* System.out.println(Math.pow(3,4));*//81.0* System.out.println(Math.random());*// [0.0-1.0)之间的随机值* }  
  
}

1. 练习随机类Random相关API。

public static void main(String[] args) {

// 创建一个随机对象

Random random = new Random() ;

// 生成一个随机的int整数

System.out.println(random.nextInt());

// 随机生成一个[0,10)范围内的随机数

System.out.println(random.nextInt(10));

// 随机生成一个float类型的值

System.out.println(random.nextFloat());

// 随机生成一个double类型的值

System.out.println(random.nextDouble());

// 随机生成一个long类型的值

System.out.println(random.nextLong());

// 随机生成一个boolean类型的值

System.out.println(random.nextBoolean());

}

1. 将Date类对象跟字符串格式互转。

将Date对象转换成自定义的格式输出。

public static void main(String[] args) throws ParseException {

// 定义当前时间点对应的Date对象

Date date = new Date();

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss") ;

//将日期Date格式化输出

String dateString = format.format(date);

System.out.println(dateString);//2020-12-28 14:57:18

}

将字符串格式的日期转成Date类型

public static void main(String[] args) throws ParseException {

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss") ;

//字符串类型的日期

String d = "20-12-28 14:55:05";

// 字符串类型的日期转成Date对象

Date date = format.parse(d);

System.out.println(date);

}