## ****任务单作答卡****

|  |  |
| --- | --- |
| ****开始时间：2022-7-19 15:20**** | ****结束时间：2022-7-19 16:20**** |
| **作者： 郭有根 修订次数： 1** | **任务状态： 正常  延迟** |

### ****一、问答题****

**1、请分别讲一下工厂模式、模板模式的作用和使用场景。**

**工厂模式：用于单纯创建某一类的对象**

**模板模式：有固定的执行流程，而且有些父类完成不了 只能子类来**

1. **请阐述浅拷贝和深拷贝的区别**

**浅拷贝只能克隆对象，对于引用的 多加一层的对象 不能复制**

**深拷贝 把要复制的和引用的对象都复制了一遍。**

**3、请说一说 toString方法的意义**

**将对象用字符串传出去 利用返回对象内存地址来实现 所以很简陋 需要重写**

### **二、编程题**

**1.设计一个客户类[Customer]，包含如下属性：**

**id**

**客户名**

**客户电话**

**客户等级**

**要求：生成构造、getter/setter、hashcode和equals以及toString方法**

**2、设计一个订单类[Order]，包含如下属性：**

**id**

**订单编号【内部生成，规则另定，此属性值由后面提供的订单号生成器类来生成】**

**订单费用**

**下单时间**

**出货时间**

**订单状态**

**所属客户[Customer]**

**要求：生成 构造、getter/setter、hashCode和equals方法，以及toString方法。注：在toString方法中不要出现**

**customer属性**

**3、为了让生成的订单号不重复，我们设计一个订单号生成器类来生成订单号：**

**订单号生成规则如下：**

**由年月日时分秒+3位数字组成，这4位数组从000开始编号，一直到999， 1秒内不能超过1000个订单。**

**在同一个秒内，这3位数字依次递增，不能重复，如果在同一秒内并发量很大，请求生成订单号超过1000个，**

**则程序需要把此秒给空耗过去，直到下一秒时，重新从000开始编号。如果编号没有超过999，即使时间进入到**

**另一个秒内，编号从上次的位置继续，不用从000开始。【算法请自行实现】**

**比如：**

**根据上面的要求，我们可以定义一个订单号生成器类：**

**public class Customer {**

**// TODO ...**

**}**

**public class Order {**

**//TODO ...**

**//这里要注意一下，订单编号 属性可以直接调用 订单号生成器类.generateNo() 来生成.**

**}**

**20200908114218000**

**20200908114218001**

**20200908114218002**

**...**

**20200908114218999**

**-- 上面是在18秒中，编号到了999，则程序需要空耗到19秒处，重新从000开始编号，如下 ：**

**20200908114219000**

**20200908114219001**

**...**

**注：如果对于LocalDateTime 不熟悉的话，则可以查看相关的API。这里给出一些代码片断，以供参考**

**最后，写一个测试类，循环生成1010个订单，并输出这1010个订单的编号，程序完成得好的话，可以感觉到当**

**编号到达999时，会有一个显明的"停顿"时间， 因为程序要进入到下一个秒。**

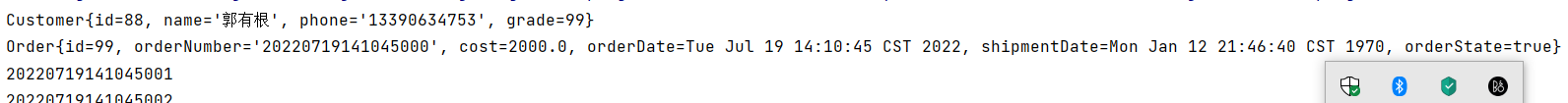
**customer类**

|  |
| --- |
| **package com.se.day9;**  **import java.util.Objects;**  **/\*\***  **\* @projectname: learn**  **\* @packname: com.se.day9**  **\* @classname: Customer**  **\* @author: gyg**  **\* @date: 2022/7/19 上午 9:57**  **\* @version: 1.0**  **\* @description: TODO**  **\*/**  **public class Customer {**  **private long id;**  **private static long idincrease=88l;//利用static来实现id自增且不重复**  **private String name;**  **private String phone;**  **private int grade;**  **public Customer(String name,String phone,int grade){**  **this.id=idincrease++;**  **this.name=name;**  **this.phone=phone;**  **this.grade=grade;**  **}**  **public Customer(){**  **}**  **@Override**  **public boolean equals(Object o) {**  **if (this == o) return true;**  **if (o == null || getClass() != o.getClass()) return false;**  **Customer customer = (Customer) o;**  **return id == customer.id && grade == customer.grade && Objects.equals(name, customer.name) && Objects.equals(phone, customer.phone);**  **}**  **@Override**  **public int hashCode() {**  **return Objects.hash(id, name, phone, grade);**  **}**  **@Override**  **public String toString() {**  **return "Customer{" +**  **"id=" + id +**  **", name='" + name + '\'' +**  **", phone='" + phone + '\'' +**  **", grade=" + grade +**  **'}';**  **}**  **public long getId() {**  **return id;**  **}**  **public void setId(long id) {**  **this.id = id;**  **}**  **public static long getIdincrease() {**  **return idincrease;**  **}**  **public static void setIdincrease(long idincrease) {**  **Customer.idincrease = idincrease;**  **}**  **public String getName() {**  **return name;**  **}**  **public void setName(String name) {**  **this.name = name;**  **}**  **public String getPhone() {**  **return phone;**  **}**  **public void setPhone(String phone) {**  **this.phone = phone;**  **}**  **public int getGrade() {**  **return grade;**  **}**  **public void setGrade(int grade) {**  **this.grade = grade;**  **}**  **}** |
|  |
| **order类** |
| **package com.se.day9;**  **import java.util.Date;**  **import java.util.Objects;**  **/\*\***  **\* @projectname: learn**  **\* @packname: com.se.day9**  **\* @classname: Order**  **\* @author: gyg**  **\* @date: 2022/7/19 上午 9:53**  **\* @version: 1.0**  **\* @description: TODO**  **\*/**  **public class Order {**  **private long id;**  **private static long increase=99l;//实现id自增**  **private String orderNumber;**  **private double cost;**  **private Date orderDate;**  **private Date shipmentDate;**  **private boolean orderState;**  **private Customer customer;**  **public Order(double cost, Date orderDate, Date shipmentDate, boolean orderState, Customer customer) {**  **this.id = increase++;**  **orderNumber=OrderNoGenerator.generateNo();**  **this.cost = cost;**  **this.orderDate = orderDate;**  **this.shipmentDate = shipmentDate;**  **this.orderState = orderState;**  **this.customer = customer;**  **}**  **@Override**  **public boolean equals(Object o) {**  **if (this == o) return true;**  **if (o == null || getClass() != o.getClass()) return false;**  **Order order = (Order) o;**  **return id == order.id && Double.compare(order.cost, cost) == 0 && orderState == order.orderState && Objects.equals(orderNumber, order.orderNumber) && Objects.equals(orderDate, order.orderDate) && Objects.equals(shipmentDate, order.shipmentDate) && Objects.equals(customer, order.customer);**  **}**  **@Override**  **public int hashCode() {**  **return Objects.hash(id, orderNumber, cost, orderDate, shipmentDate, orderState, customer);**  **}**  **@Override**  **public String toString() {**  **return "Order{" +**  **"id=" + id +**  **", orderNumber='" + orderNumber + '\'' +**  **", cost=" + cost +**  **", orderDate=" + orderDate +**  **", shipmentDate=" + shipmentDate +**  **", orderState=" + orderState +**  **'}';**  **}**  **public long getId() {**  **return id;**  **}**  **public void setId(long id) {**  **this.id = id;**  **}**  **public static long getIncrease() {**  **return increase;**  **}**  **public static void setIncrease(long increase) {**  **Order.increase = increase;**  **}**  **public String getOrderNumber() {**  **return orderNumber;**  **}**  **public void setOrderNumber(String orderNumber) {**  **this.orderNumber = orderNumber;**  **}**  **public double getCost() {**  **return cost;**  **}**  **public void setCost(double cost) {**  **this.cost = cost;**  **}**  **public Date getOrderDate() {**  **return orderDate;**  **}**  **public void setOrderDate(Date orderDate) {**  **this.orderDate = orderDate;**  **}**  **public Date getShipmentDate() {**  **return shipmentDate;**  **}**  **public void setShipmentDate(Date shipmentDate) {**  **this.shipmentDate = shipmentDate;**  **}**  **public boolean isOrderState() {**  **return orderState;**  **}**  **public void setOrderState(boolean orderState) {**  **this.orderState = orderState;**  **}**  **public Customer getCustomer() {**  **return customer;**  **}**  **public void setCustomer(Customer customer) {**  **this.customer = customer;**  **}**  **}**  **OrderNoGenerator类**  **package com.se.day9;**  **import java.time.LocalDateTime;**  **import java.time.format.DateTimeFormatter;**  **/\*\***  **\* @projectname: learn**  **\* @packname: com.se.day9**  **\* @classname: OrderNoGenerator**  **\* @author: gyg**  **\* @date: 2022/7/19 上午 10:06**  **\* @version: 1.0**  **\* @description: TODO**  **\*/**  **public class OrderNoGenerator {**  **private static int seq=0;//编号**  **private static int count=0;//一秒内记数**  **private static DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyyMMddHHmmss");**  **private static long sec=System.currentTimeMillis();//秒数**  **public static String generateNo(){**  **LocalDateTime now = LocalDateTime.now();**  **String str = dtf.format(now);**  **if((System.currentTimeMillis()-sec)<1000){//间隔在一秒之内的情况**  **if(count>=1000) {**  **count=0;//记数清空 但编号不清空**  **while((System.currentTimeMillis()-sec)<1000){}//等待 进入下一秒**  **sec=System.currentTimeMillis();**  **}**  **if(seq>=1000)**  **seq=0;**  **String iStr=String.format("%03d",seq++);**  **count++;**  **return str+iStr;**  **}**  **else { //间隔大于一秒**  **count=0;**  **sec=System.currentTimeMillis();**  **if(seq>=1000)**  **seq=0;**  **String iStr=String.format("%03d",seq++);**  **count++;**  **return str+iStr;**  **}**  **}**  **}**  **测试**  **package com.se.day9;**  **import java.util.Date;**  **/\*\***  **\* @projectname: learn**  **\* @packname: com.se.day9**  **\* @classname: Ordertest**  **\* @author: gyg**  **\* @date: 2022/7/19 上午 10:33**  **\* @version: 1.0**  **\* @description: TODO**  **\*/**  **public class Ordertest {**  **public static void main(String[] args) {**  **Customer c = new Customer("郭有根","13390634753",99);**  **System.out.println(c.toString());**  **Date orderdate =new Date();**  **Date shipmentdate=new Date(1000000000);**  **Order o = new Order(2000,orderdate,shipmentdate,true,c);**  **System.out.println(o.toString());**  **for(int i=0;i<1020;i++){**  **System.out.println(OrderNoGenerator.generateNo());**  **}**  **}**  **}** |

**运行的截图：[可以有多个]**

**如：**

**tostring方法**



在999处暂停一秒左右

