

# 暨南大学本科实验报告专用纸(附页)

---

## UID 管理器

\* 实验项目类型：设计性

\* 此表由学生按顺序填写

课程名称 面向对象程序设计/JAVA 语言 成绩评定           

实验项目名称 UID 管理器 指导老师 干晓聪

实验项目编号 1 实验项目类型 设计性 实验地点 数学系机房

学生姓名 郭彦培 学号 2022101149

学院 信息科学技术学院 系 数学系 专业 信息管理与信息系统

实验时间 2023 年 10 月 27 日上午 ~ 2023 年 10 月 27 日中午

## 一、实验目的

熟悉基础的面向对象知识，包括成员变量、方法，访问控制与构造函数

## 二、实验环境

计算机：PC X64

操作系统：Windows

编程语言：Java

IDE：Visual Studio Code

## 三、程序原理

提供三种类型的 UID 生成、申请与维护。

1. 基于日期的 UID，例如 2022101149
2. 完全无序的类激活码，例如 AA2CA-STBS3-AED2P-RDGSSP
3. 按顺序发放的序列，可选固定位数，例如 00001,00002

可以保证所有 UID 不会重复。

类中储存所有 UID 对应的引用。

# 暨南大学本科实验报告专用纸(附页)

申请复杂度  $O(1)$ ，引用复杂度  $O(\log(n))$

测试用例的控制台规则：

中括号内为需要填入字符串，尖括号为可选参数，默认为列表第一个。

<code>getUID &lt;type: "date"   "code"   "seq"&gt; [name]</code>	申请对应类型的新 UID，并与一个引用名称 <code>name</code> 绑定
<code>secUID &lt;type: "date"   "code"   "seq"&gt; [UID]</code>	查找 UID 对应的引用名称

## 四、程序代码

文件：`sis2\UIDmanager.java` 实现了 `UIDmanager` 类

```
package sis2;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Random;

public class UIDmanager {
    private HashMap<String, Object> dateHM = new HashMap<String, Object>();
    private HashMap<String, Object> codeHM = new HashMap<String, Object>();
    private HashMap<String, Object> seqHM = new HashMap<String, Object>();

    final Random rd = new Random();

    final Integer DATENUM_LENGTH;
    final Integer SEQ_LENGTH;
    final Integer CODE_LENGTH;
    private Integer nextDataNum = 0;
    private Integer nextSeq = 0;

    private char getRandChar()
    {
        int flg = rd.nextInt(26+10);
        if(flg < 10) return (char)(48+flg);
        else return (char)(55+flg);
    }
}
```

# 暨南大学本科实验报告专用纸(附页)

---

```
public UIDmanager()
{
    DATENUM_LENGTH = 4;
    SEQ_LENGTH = 10;
    CODE_LENGTH = 5;
}

public UIDmanager(int _SEQ_LENGTH,int _DATENUM_LENGTH,int _CODE_LENGTH)
{
    DATENUM_LENGTH = _DATENUM_LENGTH;
    SEQ_LENGTH = _SEQ_LENGTH;
    CODE_LENGTH = _CODE_LENGTH;
}

public String nextDate(Object c)
{
    Date date = new Date();
    SimpleDateFormat sf = new SimpleDateFormat("yyMMdd");
    String s = sf.format(date);
    nextDataNum++;
    Integer tmp = nextDataNum;
    StringBuffer sb = new StringBuffer();
    for(int i = 0;i < DATENUM_LENGTH;i ++)
    {
        sb.append(tmp % 10);
        tmp /= 10;
    }
    String rt = s + sb.reverse().toString();
    dateHM.put(rt, c);
    return rt;
}

public String nextDate()
{
    Date date = new Date();
    SimpleDateFormat sf = new SimpleDateFormat("yyMMdd");
    String s = sf.format(date);
    nextDataNum++;
    Integer tmp = nextDataNum;
    StringBuffer sb = new StringBuffer();
    for(int i = 0;i < DATENUM_LENGTH;i ++)
    {
        sb.append(tmp % 10);
```

# 暨南大学本科实验报告专用纸(附页)

---

```
        tmp /= 10;
    }
    String rt = s + sb.reverse().toString();
    return rt;
}

public String nextCode(Object c)
{
    StringBuffer rt = new StringBuffer();
    for(int k = 0; k < 4; k++)
    {
        for(int i = 0; i < CODE_LENGTH; i++)
        {
            rt.append(getRandChar());
        }
        if(k != 3)rt.append('-');
    }
    codeHM.put(rt.toString(), c);
    return rt.toString();
}

public String nextCode()
{
    StringBuffer rt = new StringBuffer();
    for(int k = 0; k < 4; k++)
    {
        for(int i = 0; i < CODE_LENGTH; i++)
        {
            rt.append(getRandChar());
        }
        if(k != 3)rt.append('-');
    }
    return rt.toString();
}

public String nextSeq(Object c)
{
    StringBuffer rt = new StringBuffer();
    nextSeq++;
    Integer tmp = nextSeq;
    for(int i = 0; i < SEQ_LENGTH; i++)
    {
        rt.append(tmp % 10);
        tmp /= 10;
    }
}
```

# 暨南大学本科实验报告专用纸(附页)

---

```
    }
    seqHM.put(rt.reverse().toString(), c);
    return rt.toString();
}

public String nextSeq()
{
    StringBuffer rt = new StringBuffer();
    nextSeq++;
    Integer tmp = nextSeq;
    for(int i = 0; i < SEQ_LENGTH; i++)
    {
        rt.append(tmp % 10);
        tmp /= 10;
    }
    return rt.reverse().toString();
}

public Object secDate(String uid) throws Exception
{
    return dateHM.get(uid);
}

public Object secCode(String uid) throws Exception
{
    return codeHM.get(uid);
}

public Object secSeq(String uid) throws Exception
{
    return seqHM.get(uid);
}

public void bindUID(String uid, Object c) throws Exception
{
    if(uid.length() == 4 + DATENUM_LENGTH) dateHM.put(uid, c);
    else if(uid.length() == 4 + CODE_LENGTH * 4) codeHM.put(uid, c);
    else if(uid.length() == SEQ_LENGTH) seqHM.put(uid, c);
    else throw new Exception("UID 格式错误\n");
}
}
```

文件: `sis2\test.java` 用于实现指令交互的测试

# 暨南大学本科实验报告专用纸(附页)

---

```
package sis2;
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        UIDmanager uidm = new UIDmanager();
        try (Scanner sc = new Scanner(System.in)) {
            for(;;)
            {
                String cmd = sc.next();
                if(cmd.equals("getUID"))
                {
                    String type = sc.next();
                    if(type.equals("date"))
                    {
                        String name = sc.next();
                        System.out.println(uidm.nextDate(name));
                    }
                    else if(type.equals("code"))
                    {
                        String name = sc.next();
                        System.out.println(uidm.nextCode(name));
                    }
                    else if(type.equals("seq"))
                    {
                        String name = sc.next();
                        System.out.println(uidm.nextSeq(name));
                    }
                }
                else if(cmd.equals("secUID"))
                {
                    String type = sc.next();
                    if(type.equals("date"))
                    {
                        String uid = sc.next();
                        System.out.println(uidm.secDate(uid));
                    }
                    else if(type.equals("code"))
                    {
                        String uid = sc.next();
                        System.out.println(uidm.secCode(uid));
                    }
                    else if(type.equals("seq"))
                    {

```

```
String uid = sc.next();
System.out.println(uidm.secSeq(uid));
    }
    } else break;
}
sc.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
```

五、 出现的问题、原因与解决方法

编码过程十分顺利

六、 测试数据与运行结果

输入	输出	解释
getUID date GYP	2312150001	生成了一个基于当前日期的 UID 并且与 GYP 绑定
getUID code program_by_GYP	7EFBG-GM1E6-8KFL9-7MU08	生成一个随机串并与 program_by_GYP 绑定
getUID seq seq1	0000000001	生成一个顺序编号的 ID 并与 seq1 绑定
getUID seq seq2	0000000002	生成一个顺序编号的 ID 并与 seq2 绑定
secUID date ??	UID 格式错误	查询 UID 时使用了错误格式
secUID date 2312150001	GYP	查询到 2312150001 所绑定的数据为 GYP
secUID code 7EFBG-GM1E6-8KFL9-7MU08	program_by_GYP	查询到 7EFBG-GM1E6-8KFL9-7MU08 所绑定的数据为 program_by_GYP