重庆科技学院

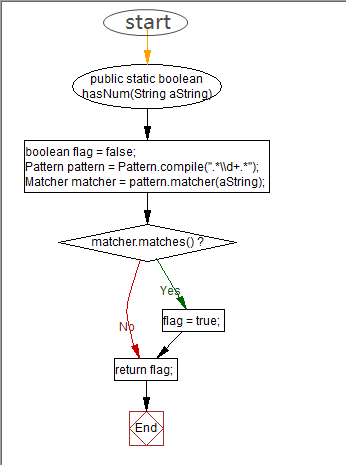
上机实验报告(程序设计类)

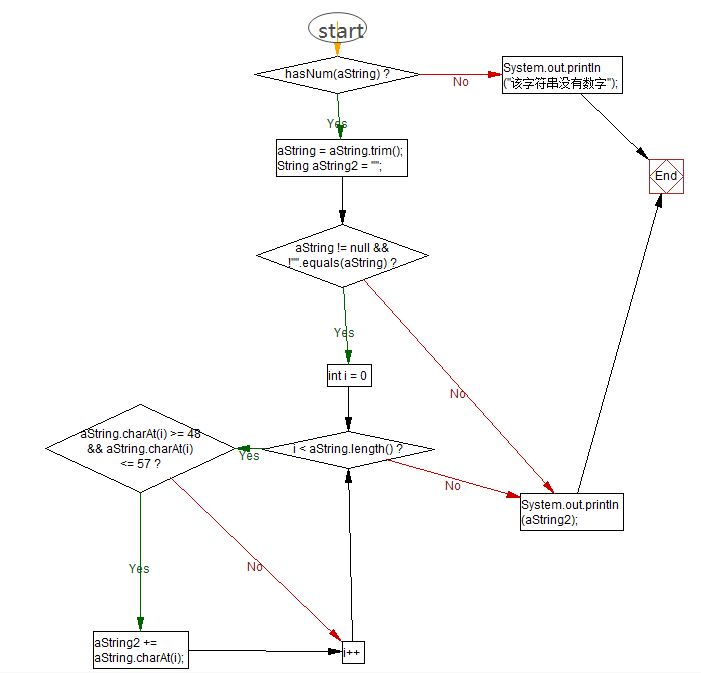
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 面向对象的序设计方法 | | 实验题目 | 实验三Java 基本类库和集合框架的使用 | | |
| 机房编号 | I301 | | 上机时间 | 2018-10-29 | | |
| 指导老师 | 王双明 | | 报告序号 | 3 | 上机成绩 | . |
| 学生姓名 | 袁阊越 | 学号 | 2018520612 | 教学班级 | 计科升2016 | |
| **一、上机目的和要求:**   1. 掌握 String 类和 StringBuffer 类的使用； 2. 掌握 Math 和 Random 类的使用； 3. 掌握包装类的使用； 4. 掌握日期和时间的处理方法； 5. 掌握集合 List、Set 和 Map 的使用； 6. 掌握 Iterator 迭代器的使用，掌握 for…each 循环的使用； 7. 了解泛型的作用以及使用； 8. 掌握 String 型与基本数据类型的转换、String 型与日期型的转换。 9. 掌握 Scanner 类的使用 | | | | | | |
| **二、程序开发环境:**  处理器：Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz (8 CPUs), ~3.4GHz  操作系统：Windows 7 专业版 32-bit  内存：4GB  JDK：jdk1.6.0\_04  软件：eclipse | | | | | | |
| **三、上机内容(老师布置的具体任务):**   1. 编写程序剔除一个字符串中的全部非数字字符，例如：将形如“A1BC2you3”中的非数字字符全部剔除，得到字符串“123”，并将“123”转换为 int 型并输出，如果输入字符串中不含数字，则输出“无数字”。 2. 编写程序将 18 位的身份号证号码中表示年份的信息显示为\*，其余的字符保持不变（例如：将身份证号码 10013319961213602X 转换为 100133\*\*\*\*1213602X 显示）。   3．编写程序实现，大小写字母的切换，其他字符不变（例如：输入字符串“！”则输出为“i LOVE jAVA！”）。   1. 运用面向对象的程序设计思想，把表 1 中的学生信息存储在 List 集合（ArrayList 或者LinkedList）中。编写程序实现下列功能：   （1）创建 Student 学生类，类中包括三个成员变量，分别为学号、姓名、出生日期，添加不带参数的构造方法，添加含以上三个参数的构造方法，添加 Getter 和 Setter 方法，    复写 toString()方法（返回字符串格式：学号 姓名 出生日期）  （2）将表 1 中的学生依次添加到 List 集合中；  yue  （3）输出所有的学生信息（最好使用迭代器 Iterator）；  （4）有学生（学号：201644008，姓名：李明，出生日期：1991 年 1 月 1 日）中途入学，将其添加到 List 集合中；  （5）有学生王五中途退学，将其从 List 集合中删除；  （6）再一次遍历输出所有的学生信息（最好使用 foreach 语句实现）。  （7）输出年龄最小的学生的学号、姓名和出生日期（考虑最小年龄的学生不止一个的情况）   1. **（选做）**编写程序，产生 1 个随机电话号码，电话号码的前五位是 15923，后六位随机产生。 2. **（选做）**编写程序，随机产生一个 4 位的验证码，验证码由 26 位英文大写或者小写字符构成。 3. **（选做）**编写程序，实现输入银行卡号，以每 4 个字符加一个空格显示。例如，假如银行卡号是 62220630108589564785130，则输出为 6222 0630 1085 8956 4785 130。 4. **（选做）**编写程序，如果输入一个英文句子，将每个单词的首字符改成大写后再输   出。 | | | | | | |
| **四、上机调试中出现的错误信息、错误原因及解决办法:**  错误信息:java.lang.IndexOutOfBoundsException: Index: 5, Size: 5  原因： 超出数组长度  解决方法：循环-1。 | | | | | | |
| **五、上机实验中的收获及心得:**  在做Java类和对象的设计的实验前，我以为不会难做，就像以前做物理实验一样，做完实验，然后两下子就将实验报告做完。直到做完测试实验时，我才明白其实并不容易做，但学到的知识与难度成正比，使我受益匪浅。  　　在做实验的过程中我还学会了 String 类和 StringBuffer 类的使用，掌握 Math 和 Random 类的使用，包装类的使用，日期和时间的处理方法，集合 List、Set 和 Map 的使用，Iterator 迭代器的使用，掌握 for…each 循环的使用；String 型与基本数据类型的转换、String 型与日期型的转换。 Scanner 类的使用。  　　透过这次Java类和对象的设计的实验，使我学到了不少实用的知识，更重要的是，做实验的过程，思考问题的方法，这与做其他的实验是通用的，真正使我们受益匪浅。 | | | | | | |

建议：正文采用五宋体，代码用小五号Times new Roman字体

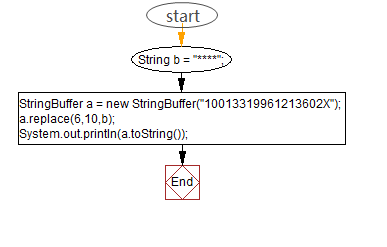
附录：

**一、程序流程图**

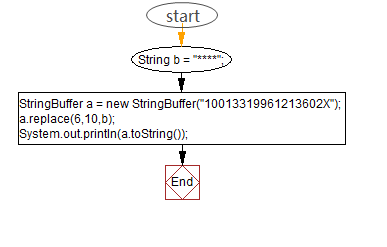




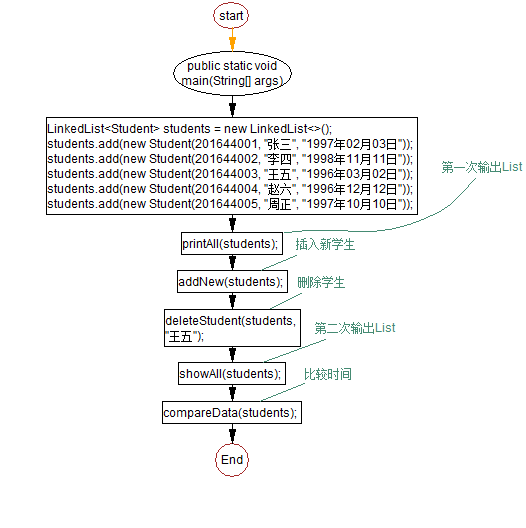
第一题

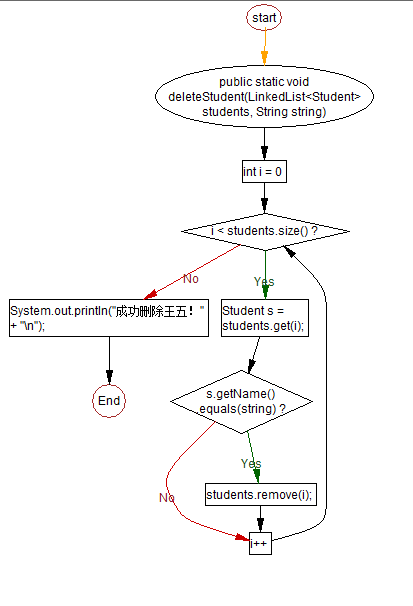
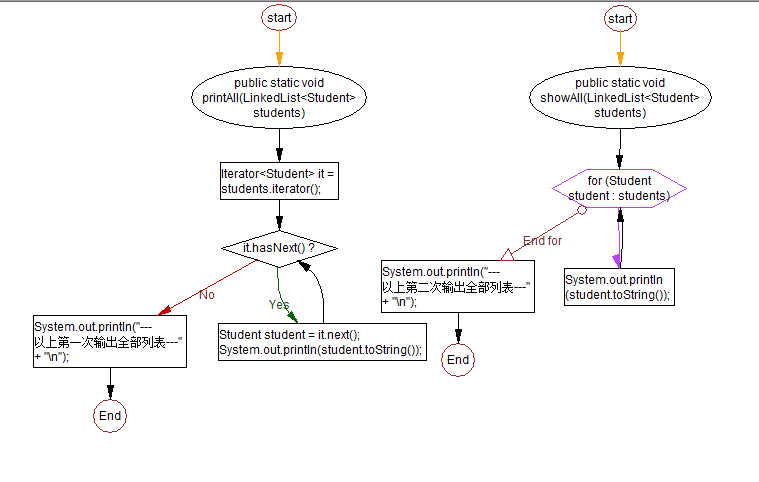


第二题

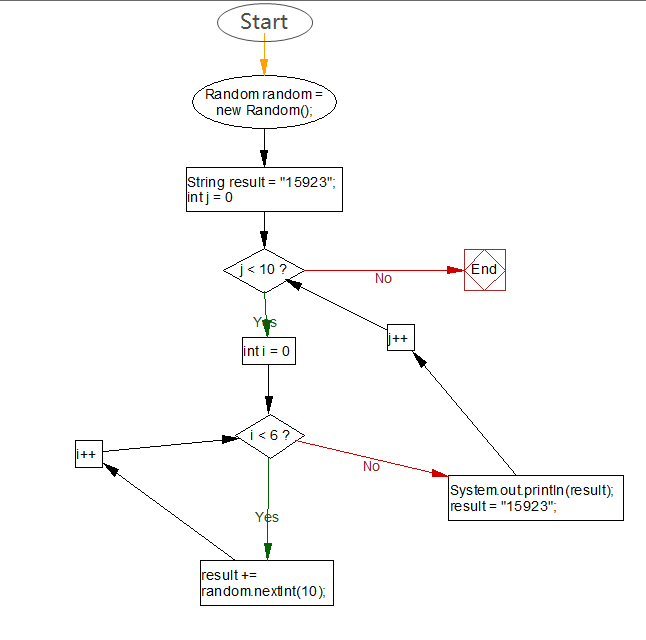


第三题

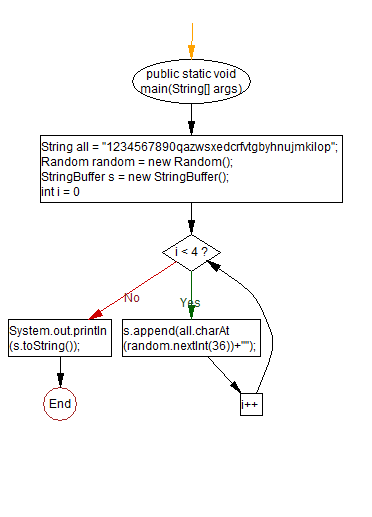




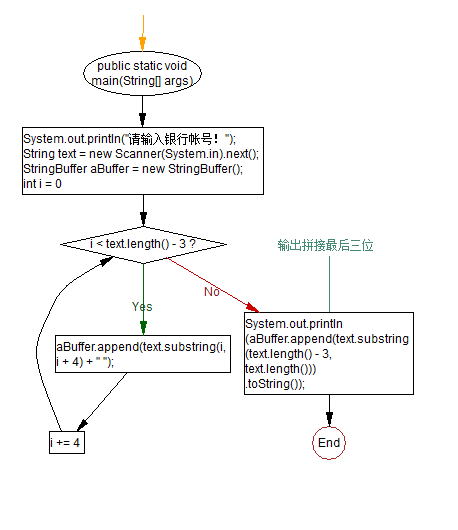
第四题



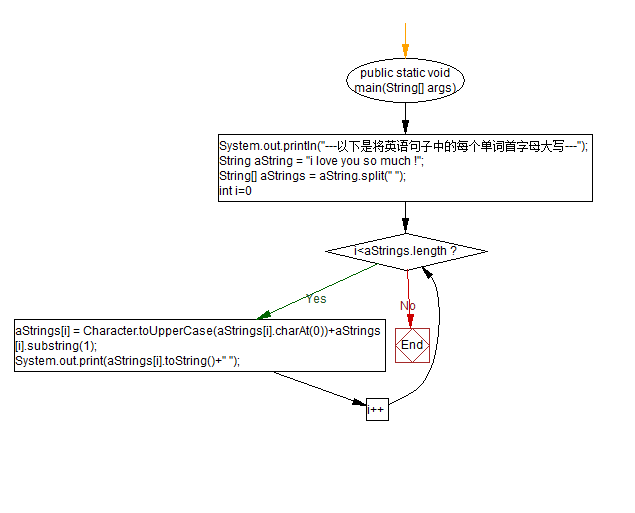
第五题



第六题



第七题



第八题

**二、主要数据结构和变量说明**

**String：**字符串广泛应用 在Java 编程中，在 Java 中字符串属于对象，Java 提供了 String 类来创建和操作字符串。

**Int：**int 数据类型是32位、有符号的以二进制补码表示的整数,一般地整型变量默认为 int 类型。

**Double：**double 数据类型是双精度、64 位、符合IEEE 754标准的浮点数，浮点数的默认类型为double类型，double类型同样不能表示精确的值，如货币.

1. **上机调试后的源程序及注释**

**public** **class** Program1 {

@SuppressWarnings("resource")

**public** **static** **void** main(String[] args) {

getNum(**new** Scanner(System.in).next());

}

// 获取字符串中的数字

**private** **static** **void** getNum(String aString) {

**if** (hasNum(aString)) {

aString = aString.trim();

String aString2 = "";

**if** (aString != **null** && !"".equals(aString)) {

**for** (**int** i = 0; i < aString.length(); i++) {

**if** (aString.charAt(i) >= 48 && aString.charAt(i) <= 57) {

aString2 += aString.charAt(i);

}

}

}

System.out.println(aString2);

} **else** {

System.out.println("该字符串没有数字");

}

}

// 判断是否存在数字，利用正则表达式。

**public** **static** **boolean** hasNum(String aString) {

**boolean** flag = **false**;

Pattern pattern = Pattern.compile(".\*\\d+.\*");

Matcher matcher = pattern.matcher(aString);

**if** (matcher.matches()) {

flag = **true**;

}

**return** flag;

}

}

**public** **class** Program2 {

**public** **static** **void** main(String[] args) {

String b = "\*\*\*\*";

StringBuffer a = **new** StringBuffer("10013319961213602X");

a.replace(6,10,b);

System.out.println(a.toString());

}

}

**public** **class** Program3 {

**public** **static** **void** main(String[] args) {

String str = "I love Java";

**char**[] ch = str.toCharArray();

StringBuffer buffer = **new** StringBuffer();

**for** (**int** i = 0; i < str.length(); i++) {

**if** (ch[i] >= 'A' && ch[i] <= 'Z') {

buffer.append(String.valueOf(ch[i]).toLowerCase());

} **else** **if** (ch[i] >= 'a' && ch[i] <= 'z') {

buffer.append(String.valueOf(ch[i]).toUpperCase());

}

}

System.out.println(buffer.toString());

}

}

Program4

**public** **class** Student {

**private** **int** ID;

**private** String name;

**private** String brith;

**public** Student(**int** iD, String name, String brith) {

**super**();

ID = iD;

**this**.name = name;

**this**.brith = brith;

}

**public** Student() {

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getID() {

**return** ID;

}

**public** **void** setID(**int** iD) {

ID = iD;

}

**public** String getBrith() {

**return** brith;

}

**public** **void** setBrith(String brith) {

**this**.brith = brith;

}

@Override

**public** String toString() {

**return** ID + " " + name + " " + brith;

}

**public** **class** Example {

**public** **static** **void** main(String[] args) {

LinkedList<Student> students = **new** LinkedList<>();

students.add(**new** Student(201644001, "张三", "1997年02月03日"));

students.add(**new** Student(201644002, "李四", "1998年11月11日"));

students.add(**new** Student(201644003, "王五", "1996年03月02日"));

students.add(**new** Student(201644004, "赵六", "1996年12月12日"));

students.add(**new** Student(201644005, "周正", "1997年10月10日"));

// 第一次输出List

printAll(students);

// 插入新学生

addNew(students);

// 删除学生

deleteStudent(students, "王五");

// 第二次输出List

showAll(students);

// 比较时间

compareData(students);

}

**public** **static** **void** addNew(LinkedList<Student> students) {

students.add(**new** Student(201644008, "李明", "1991年01月01日"));

System.out.println("---成功添加李明！---" + "\n");

}

**public** **static** **void** compareData(LinkedList<Student> students) {

SimpleDateFormat format = **new** SimpleDateFormat("yyyy年MM月dd日");

**int** id = 0;

**for** (**int** i = 0; i < students.size() - 1; i++) {

**try** {

**if** (format.parse(students.get(i).getBrith()).getTime() <= format.parse(students.get(i + 1).getBrith())

.getTime()) {

id = i - 1;

}

} **catch** (ParseException e) {

e.printStackTrace();

}

}

System.out.println("---年龄最小的学生的信息如下---");

System.out.println(students.get(id).toString());

}

**public** **static** **void** deleteStudent(LinkedList<Student> students, String string) {

**for** (**int** i = 0; i < students.size(); i++) {

Student s = students.get(i);

**if** (s.getName().equals(string)) {

students.remove(i);

}

}

System.out.println("---成功删除王五！---" + "\n");

}

**public** **static** **void** printAll(LinkedList<Student> students) {

Iterator<Student> it = students.iterator();

**while** (it.hasNext()) {

Student student = it.next();

System.out.println(student.toString());

}

System.out.println("---以上第一次输出全部列表---" + "\n");

}

**public** **static** **void** showAll(LinkedList<Student> students) {

**for** (Student student : students) {

System.out.println(student.toString());

}

System.out.println("---以上第二次输出全部列表---" + "\n");

}

}

**public** **class** Program5 {

**public** **static** **void** main(String[] args) {

Random random = **new** Random();

String result = "15923";

**for** (**int** j = 0; j < 10; j++) {

**for** (**int** i = 0; i < 6; i++) {

result += random.nextInt(10);

}

System.out.println(result);

result = "15923";

}

}

}

**public** **class** Program6 {

**public** **static** **void** main(String[] args) {

String all = "1234567890qazwsxedcrfvtgbyhnujmkilop";

Random random = **new** Random();

StringBuffer s = **new** StringBuffer();

**for** (**int** i = 0; i < 4; i++) {

s.append(all.charAt(random.nextInt(36))+"");

}

System.out.println(s.toString());

}、

**public** **class** Program7 {

**public** **static** **void** main(String[] args) {

System.***out***.println("请输入银行帐号！");

String text = **new** Scanner(System.***in***).next();

StringBuffer aBuffer = **new** StringBuffer();

**for** (**int** i = 0; i < text.length() - 3; i += 4) {

aBuffer.append(text.substring(i, i + 4) + " ");

}

//输出拼接最后三位

System.***out***.println(aBuffer.append(text.substring(text.length() - 3, text.length())).toString());

}

}

}

**public** **class** Program8 {

**public** **static** **void** main(String[] args) {

System.out.println("---以下是将英语句子中的每个单词首字母大写---");

String aString = "i love you so much !";

String[] aStrings = aString.split(" ");

**for**(**int** i=0;i<aStrings.length;i++){

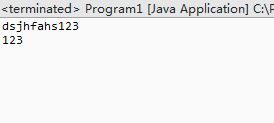
aStrings[i] = Character.toUpperCase(aStrings[i].charAt(0))+aStrings[i].substring(1);

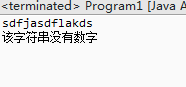
System.out.print(aStrings[i].toString()+" ");

}

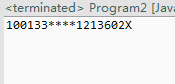
}

**四、程序运行结果截图及还存在的问题**

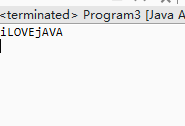




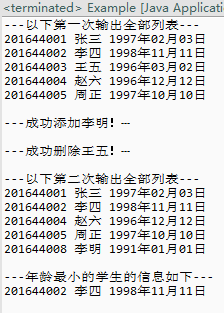
运行结果1



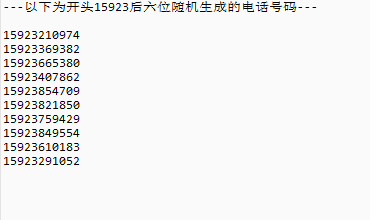
运行结果2



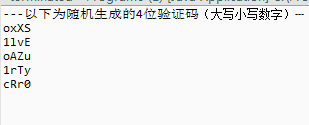
运行结果3



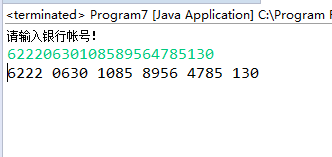
运行结果4



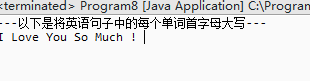
运行结果5



运行结果6



运行结果7



运行结果8