

# java第一次上机报告

## 实验目的

了解java，配置jdk，编写一些简单的java命令行程序。

## 实验题目

### 1. Class SumOfArgs

Create a class named SumOfArgs that will print out the sum of all integer arguments found on the command line. It should have a main() method so the class can be run. Anything that is not an integer should be skipped (should not contribute to the sum). It should print out only the sum (nothing else!). No error messages can be printed out (no matter what the command line args look like). Examples of what your class should do when run:

```
> java SumOfArgs
0
> java SumOfArgs 10 9 8
27
> java SumOfArgs hello dave 1 2 3
6
> java SumOfArgs Hello World
0
```

Note: Integer.parseInt(String s) parses the string argument containing the int representation as a signed decimal integer.

### 2. Class Book, Course, Student

Create three classes named Book, Course and Student to perform simple course-selecting system. Student has at least three overloaded constructors with different parameter list. Student also has some members of Book.

Examples:

```
> java Course 13131001 Java
14131001 choose Java
> java Course 14131001 Java WebEngineering
14131001 choose Java and WebEngineering
```

Notes: You really need to add more information inside each class to get this working!

## 实验内容分析

- 对于第一个题目，要求通过命令行传递参数，所以利用args数组，和c++有些区别的是，args数组每一个单元都存储参数。

主要思路就是利用java提供的autoboxing和autounboxing，这里直接用了Integer类的静态方法parseInt()把String转换成int。

这里有个问题在于传递的参数可能不是数字，而是字母之类的，所以需要判断。我想着用正则表达式，通过匹配每一个参数来决定是否累加它。

```
Pattern pat=Pattern.compile("[0-9]*");
Matcher is=pat.matcher(s);
return is.matches();
```

核心判断语句如上，使用正则表达式必须先编译它，这和其它语言一样，一次编译可以被多个Matcher对象使用。Matcher类用于保存匹配的状态，通过matches()方法判断是否匹配，只有完全匹配才会返回true，否则返回false。至于其它匹配形式文档中还有很多。

其实这里直接用String的matches(String regex)方法也行。

- 对于第二个题目简单的定义三个类即可，由于没有其它要求，如果仅仅按照题目示例看，完全可以不保存直接输出。当然，我这里还是稍稍保存了一下~~~

题目中要求至少重载三个构造函数，根据示例输入看，无非是后面课程有多。我这里直接用了个可变参数的构造函数即可满足题意（当然，这个也可以转换成三个构造函数）。

```
public Student(String...args)
{
    this.num=args[0];
    System.out.print(num+" choose ");
    for(int i=1;i<args.length-1;i++)
    {
        this.books[i]=new Book(args[i]);
        System.out.print(args[i]+" and ");
    }
}
```

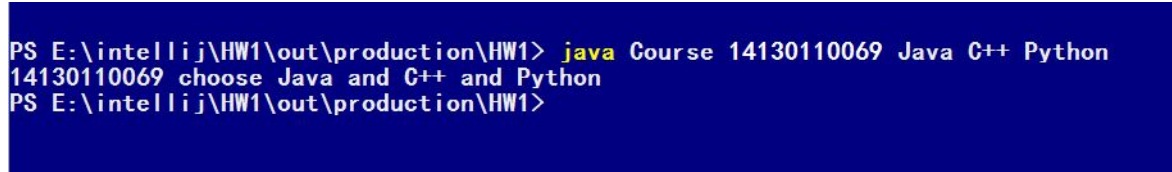
```
System.out.println(args[args.length-1]);  
this.books[args.length-1]=new Book(args[args.length-1]);  
}
```

这里直接输出并保存了，虽然在这个题中结果一样，完全不需要保存，毕竟没有写入文件中。

## 实验结果

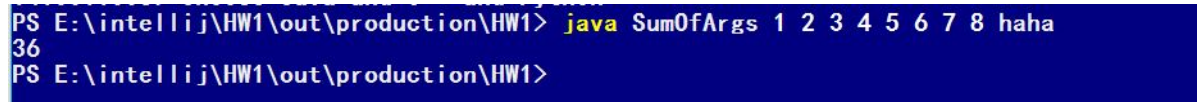
---

1. 截图如下：



```
PS E:\intellij\HW1\out\production\HW1> java Course 14130110069 Java C++ Python  
14130110069 choose Java and C++ and Python  
PS E:\intellij\HW1\out\production\HW1>
```

2. 截图如下：



```
PS E:\intellij\HW1\out\production\HW1> java SumOfArgs 1 2 3 4 5 6 7 8 haha  
36  
PS E:\intellij\HW1\out\production\HW1>
```