



**2005年1月5号**  
**Yuran Lu**



## 流程:

---

### ■ 入门指南:

联通服务器

文件使用

指令设计

工具

Maslap API

### ■ 设计原则:

### ■ Java的使用:



## 在服务器上:

---

### ■ 把这些代码放在你设定的环境中:


加入: 6.186

加入: `-f java_v1.5.0`

```
setenv JAVA_HOME /mit/java_v1.5.0
```

```
setenv CLASSPATH /mit/6.186/2005/maslab.jar:.
```

■ 如果你对服务器不太了解, 请直接找人帮助你, 那样你可以学的更快, 而且你会为自己的成就感到高兴。



## 文件的使用：

---

- Maslab API
- Java 1.5.0 API
- Sun's Java Tutorial
- Ed Faulkner's Java Reference
- All linked from SoftwareInfo page on wiki

# 程序设计

- 打开一个文字编辑器去编辑代码文件:

*emacs MyExample.java*

- 写一个详细的课堂说明，详细的阐述下你的观点和新的见解。

- 编写源代码。

- 编译:

*javac MyExample.java*


如果成功的话就生成MyExample.class

- 修正编译过程和发现其中的错误，直到成功。

- 运行:

*java MyExample*

这个是查找CLASSPATH路径下的MyExample.class程序并执行



## 工具:

---

- CVS, subversion

- make, ant

- OrcSpy, BotClient

- Instructions all on SoftwareInfo page of wiki



# The Maslab API

---

■ **maslab.orc**

■ **maslab.camera**

■ **maslab.telemetry.channel**



## 流程:

---

- 开始

- 设计法则

  - 原理

  - 模块化和设计进程

  - 写出好的说明书

  - 测试

  - 好的设计经验

- java套用



## 设计原则和目的

---

- 编译一个maslab机器人是一个需要很多人的工程
- 尽量使调试过程简单
- 必须确定不同的团队编译的代码可以合在一起运行
- 必须确定一个团队的代码的改变不影响其他团队的代码



# 模块化和设计过程

---

## ■ 模块化设计

- 提供抽象曲线
- 提供很好的控制能力，但是又要让代码便于操控。

## ■ 设计流程

- 自上而下和自下而上
- 对每个模块写出详细的规格
- 为每个模块写出代码
- 按写的测试每个模块的代码
- 测试整个系统的稳定性



# 模块化-实例

---

- 从最基本的模块开始：
- 建立更多复杂化的程序：
- 建立起高级行为：



## 写出好的规程：

---

### ■规程中应该包括的：

- 分类概要和方法
- 有多少种方法可以调用
- 对变量值的限制
- 返回值调用的方法

### ■不应该写入规程的：


- 代码是如何被执行的
- 大段的文字



## 调试:

---


- 单独测试每一段程序
- 测试系统
- 测试特别的案例
- 在编码前先测试特别的案例，或者选另一个小组成员进行测试
- 使用主要的方法
- 整体测试



## 好的设计实践：

---

- 特别重要的
- 坚持不懈的测试
- 从小开始逐渐
- 模块化
- 避免过分抽象
- 备份代码
  - 保留多个备份
  - 机器人电脑内容要另外单独备份



## 流程:

---


- 开始
- 设计原则
- **Java**中的线程
  - 动机
  - 使用线程
  - 同步



## 线程:

---

- 并行执行程序的能力
- 如果使用的得当可以让你的自动装置运行的更快
- 不同的线索和方案:
  - 图象的获得和处理
  - 保持一个最新的地图
  - 控制电流的运动
  - 高水准的控制



## 使用线程：

---

- 看**sun**公司的**java**讲座或者是Ed
- Faulkner's的java参考。
- 要特别注意避免死锁。



## 同步调节：

---

- 要注意使所有的程序互相之间不冲突。
- 同时在两边写程序可能会导致你的程序中断。