介绍

- 目标 - 当我们已经完成了这个介绍计算，你应该能够：

§ 了解软件及其基础知识
  关系到硬件

§ 编写简单的Java程序

§ 编辑，编译，并使用jGRASP运行Java程序

§ 设置断点，并通过你在程序步骤
  调试模式

§ 使用Javadoc注释在你的程序

§ 运行的Checkstyle来验证您的意见和格式

§ 为您的程序文件

背景

- 电脑系统
  - § 硬件和软件
- 硬件
  - § “物理”处理器，内存，I / O设备，...
- 软件
  - § “抽象”的指令和数据电子存储
  - § 程序指令是人类可读的文本和
    机器可读作为可执行的二进制
- 计算
  - § “的行为” - 软件运行（执行）上
    硬件，处理输入和产生输出来解决一个问题，娱乐，通信等。

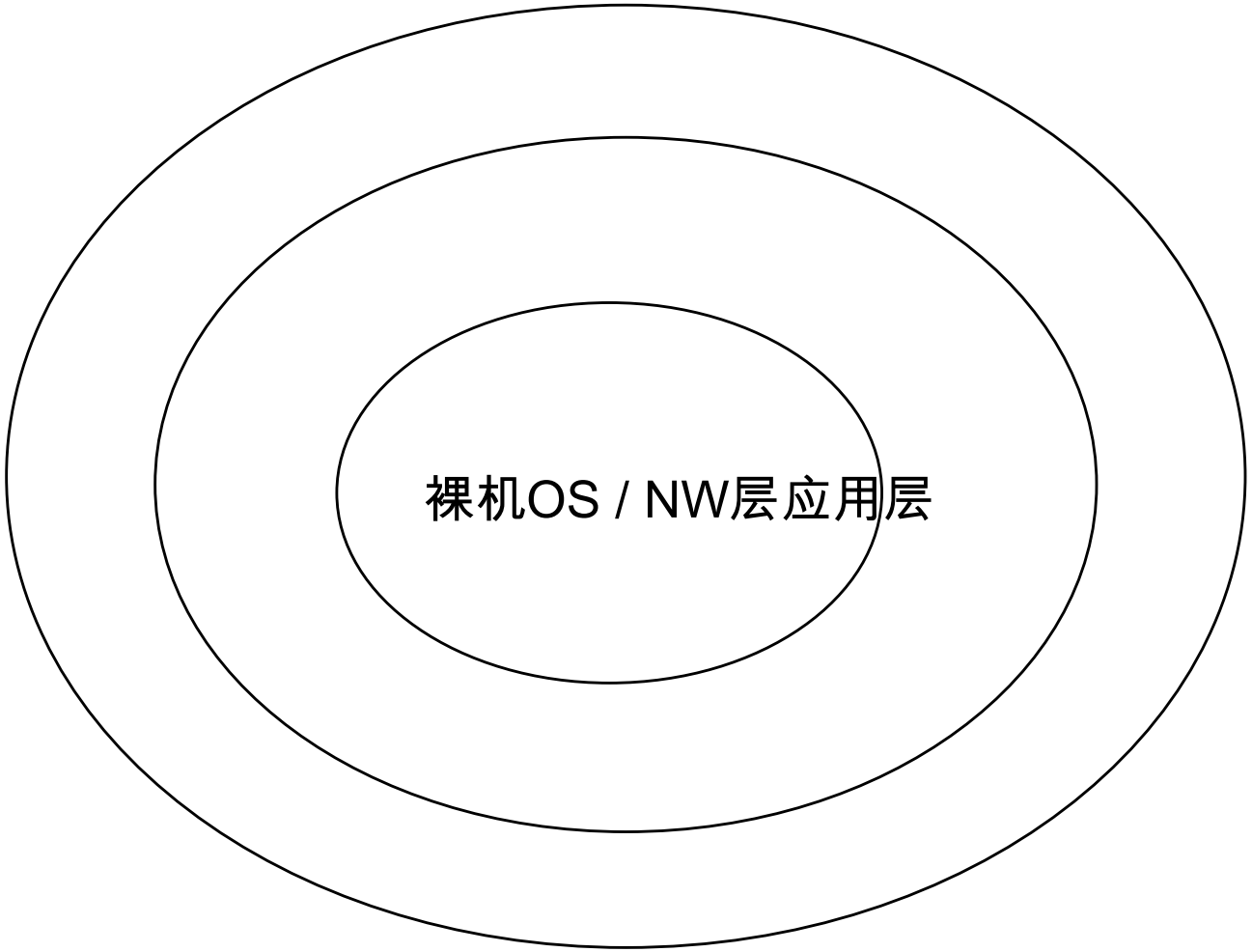
- 计算领域/学科
  - § CS + SWE（含WRSwE）+ CPE + IS + IT + ...

软件和硬件的关系

- 裸机
  - § 所有的物理组件，设备，微码
- OS /网络层
  - § 所有系统软件：OS，网络，设备驱动程序
    （在Windows，Linux，MacOS的，UNIX）
  - § 所有硬件的管理：处理器，内存，
    I / O设备
  - § 所有正在运行的软件管理（多
    流程）
- 应用层
  - § 所有的应用软件：微软Office，互联网
    浏览器的IDE（集成开发环境），编译器，...，其中包括

    写在这个课程方案

软件和硬件的关系



软件

- 在这个过程中
  - § HW假设; 设计/由CPE，EE实现
    物理学家等。
  - § SW是我们的重点; 设计/ CS通过，瑞典实施，
    IS等。
- 开发SW为约
  - § 解决问题
  - § 设计，施工，测试，...
  - § 管理固有的复杂性
  - § 组织 算法 （指令）和 数据 如
    类和对象在面向对象的编程

面向对象的概念

- 类
- 对象
- 封装
- 遗产
- 多态性
- 异常处理

所有这些面向对象的概念是直接支持Java编程语言

Java的

- 一个 *程序设计语言* 指定我们可以用它来编写程序的文字和符号

  § 采用一组规则（ *句法* ）该规定如何

  文字和符号可以放在一起，形成有效

  程序语句

  § 定义的意思（语义） *程序*

  声明

- Java是由Sun Microsystems创建并介绍了1995年（被Oracle收购，2010年）

- Java的不断发展和日益重要的软件产业

Java程序结构

- 在Java编程语言：
  - § 一个 ***程序*** 由一个或多个 *类*
  - § 一个 ***类*** 包含零点或更多的数据和/或 *方法*
  - § 一个 ***方法*** 包含零个或更多的本地数据和/或程序 *声明* 这种形式的 *算法*
- 这些条款详细整个过程中探索
- 一个Java应用程序已调用包含方法的类 **主要**

第一个程序与jGRASP

1。 启动jGRASP

2。 打开一个新的文件

3。 输入程序（增量：步骤3-6）

- 该计划应打印“战鹰”三次

4。 保存程序

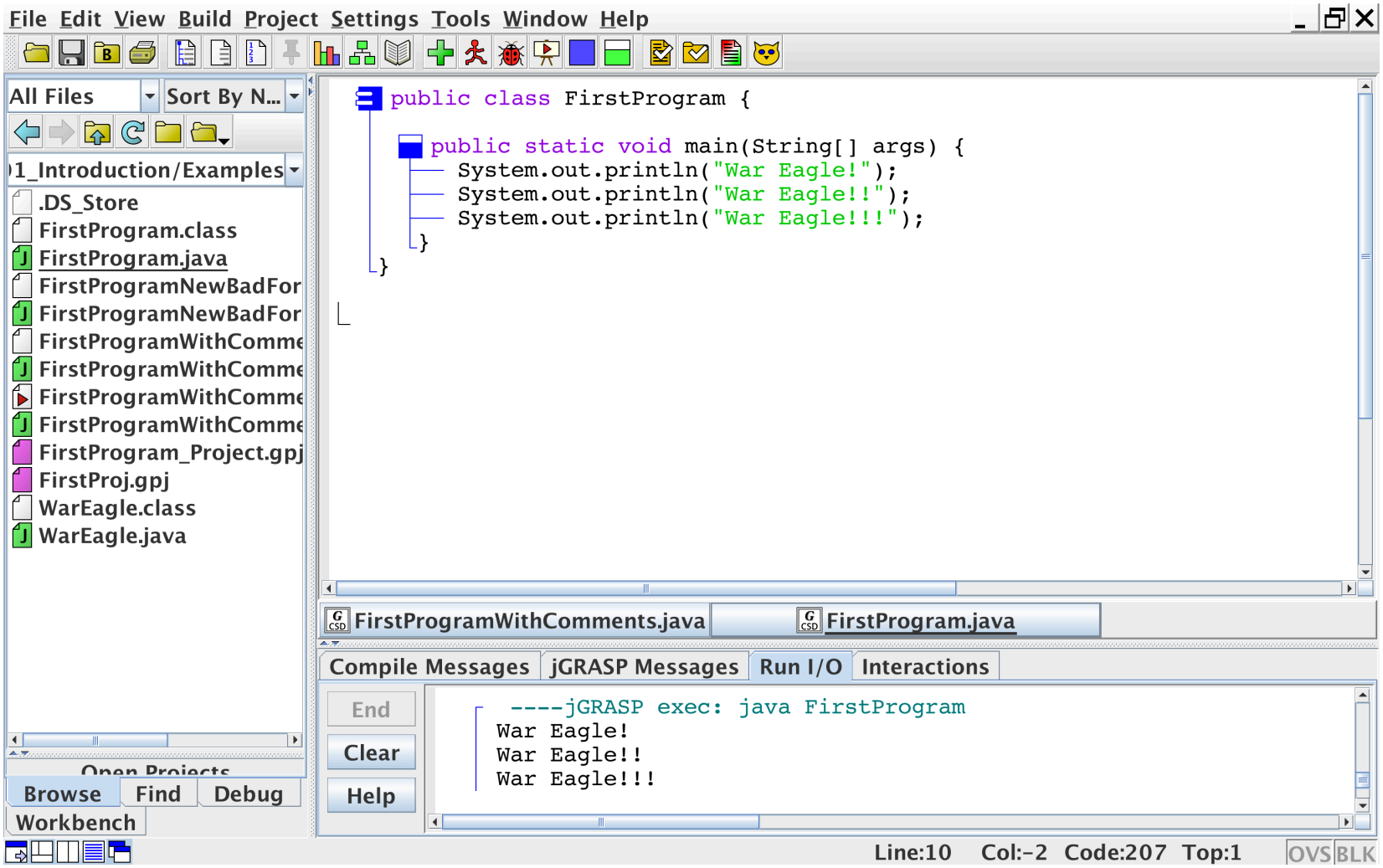
5。 编译程序

6。 运行程序（检查是否有正确的输出）

7。 设置断点和调试（步
每个语句

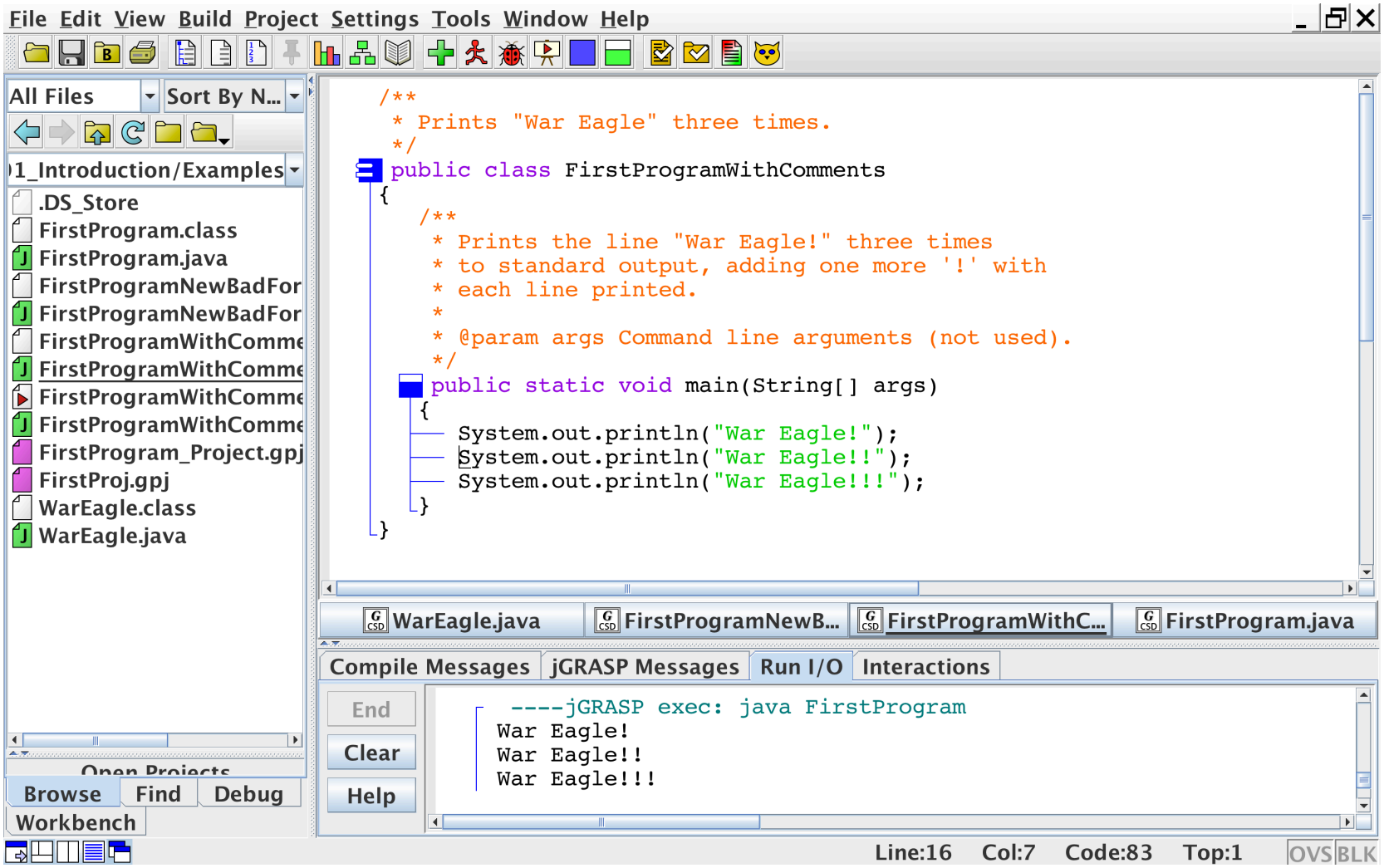
8。 生成控制结构图（CSD）
和文档; 接通/断开行号

jGRASP



FirstProgram.java

jGRASP



FirstProgramWithComments.java

软件概念

算法和数据剖析一个Java程序

程序开发，翻译，和执行

语法，语义和错误编程语言概述

面向对象编程

算法与数据

- SW :: =算法（“指令”）和数据
- 算法:: =顺序，选择指令迭代

- 伪代码（初始PROG。设计）成为“正式”的程序（即在编程语言如Java代码）

  § 伪代码可以成为该计划的意见
- 很多代码段的算法和数据
- 组织成定义对象类（面向对象编程）

剖析一个Java程序

```
/ **
  *  打印线“战鹰”！三次
  *  到标准输出。
  *
  *  @author詹姆斯·克罗斯
  *  @版 例如，日期写
  * /
公共类 FirstProgram { / **


     *  版画“战争鹰！”三次。
     *
     *  @参数ARGS的命令行参数（未使用）。
     * /
    公共静态无效 主（字串[] args）{的System.out.println（“战争鹰！”）;


        的System.out.println（“战争鹰！”）;
        的System.out.println（“战鹰!!!”）;
    }}
```

该计划的部分

- 注释
- 类
- **主要** 方法
- 身份标识
  - § 保留字
  - § 其它（如，方法和变量名）
- 的Java API
- 字面
- 空白

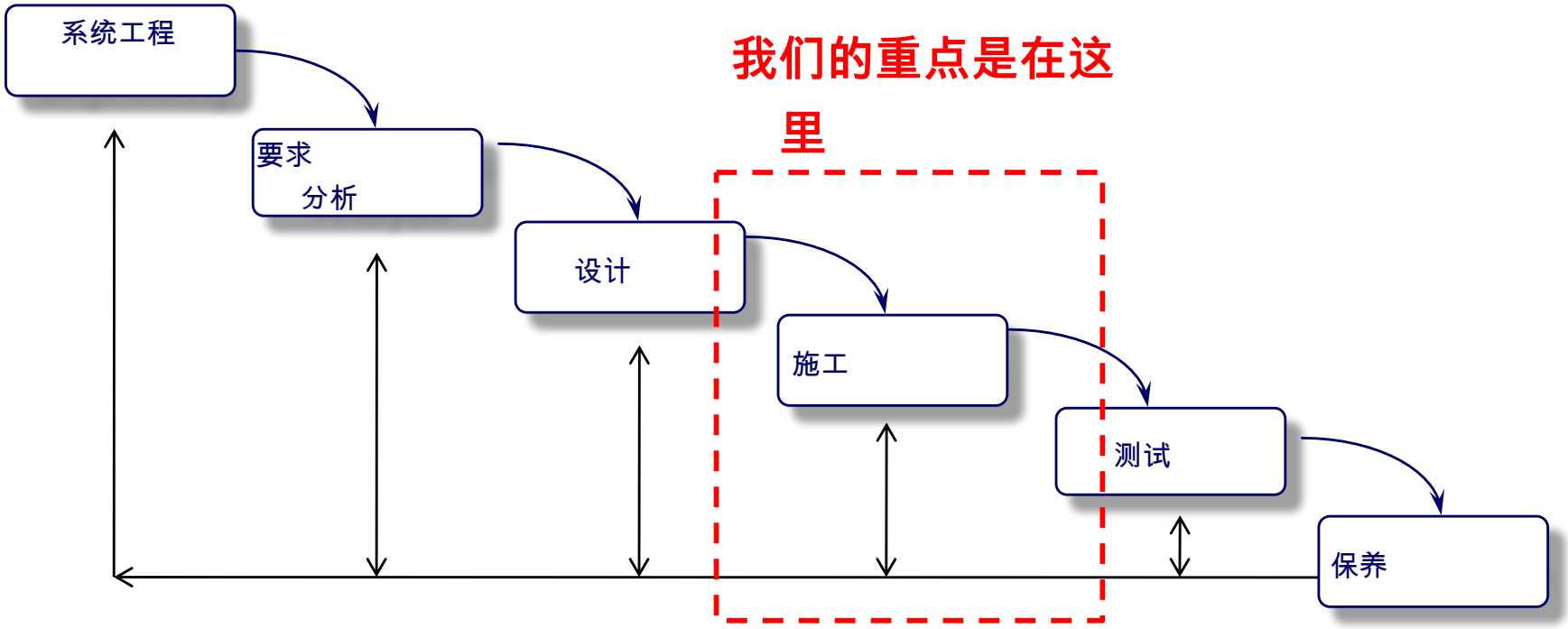
身份标识 可以是字母，数字，美元符号（$）和下划线（_）字符的任意组合; 不能以数字开头。Java是“区分大小写”。

```
/ **
  *   打印线“战鹰”！三次
  *   到标准输出。
  *
  *   @author詹姆斯·克罗斯
  * @version 例如，日期写
  * /
公共类 FirstProgram { / **

      *   版画“战争鹰！”三次。
      *
      *   @参数ARGS的命令行参数（未使用）。
      * /
     公共静态无效 主（字串[] args）{   的System.out.println（“战争鹰！

     ”）;
         的System.out.println（“战争鹰！”）;
         的System.out.println（“战鹰!!!”）;
     }}
```

Q1，Q2，Q3

程序开发

- 还有更多的软件开发比编码（又名建设方案或实施）



- 流程模型的许多变体

程序开发（续）

建筑 - 包括编码和单元测试

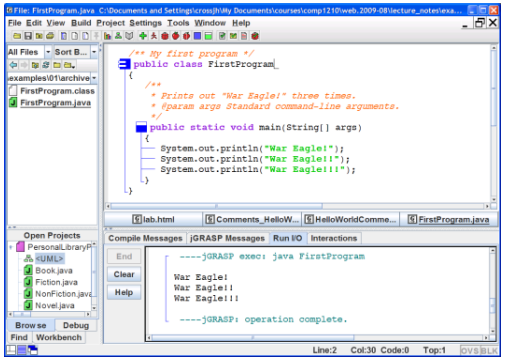
- 码
  - § 将被编译成一个编写源代码
    可执行程序。
  - § **编码标准：** 规则如何源代码
    应格式化并记录在案 - 使代码更易于阅读和调试。

- 测试（单元测试）
  - § 一旦你写你的程序，确保了
    实际输出 你的程序相匹配的
    预期输出 作为需求文档中指定。

程序开发（续）

- 程序开发工具的过程中有价值的辅助工具。

§ 一个良好的IDE（集成开发环境）与程序编辑器，调试器，互动等将要成为你最好的软件工具之一。

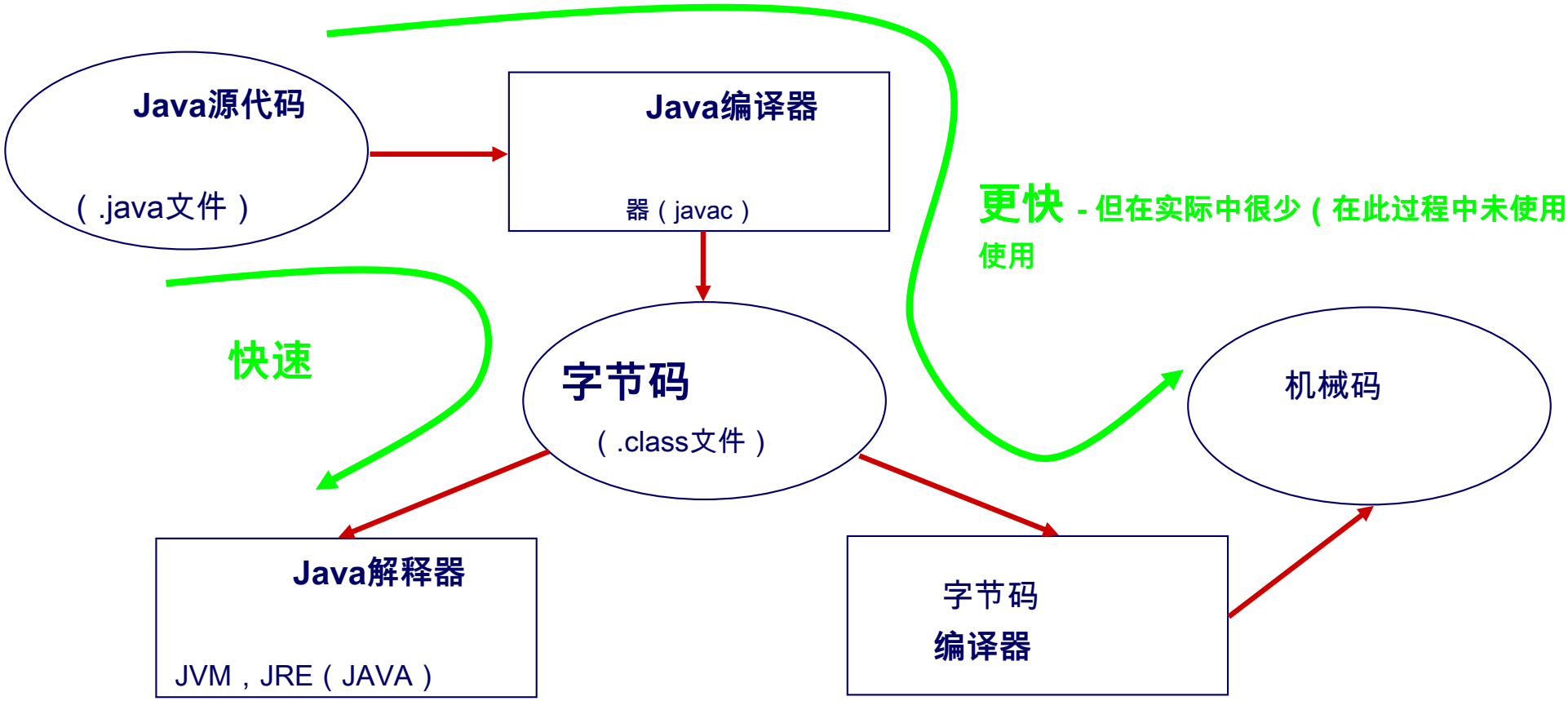


§ *jGRASP*（jgrasp.org）与Java中，Checkstyle的，JUnit中，网络-CAT

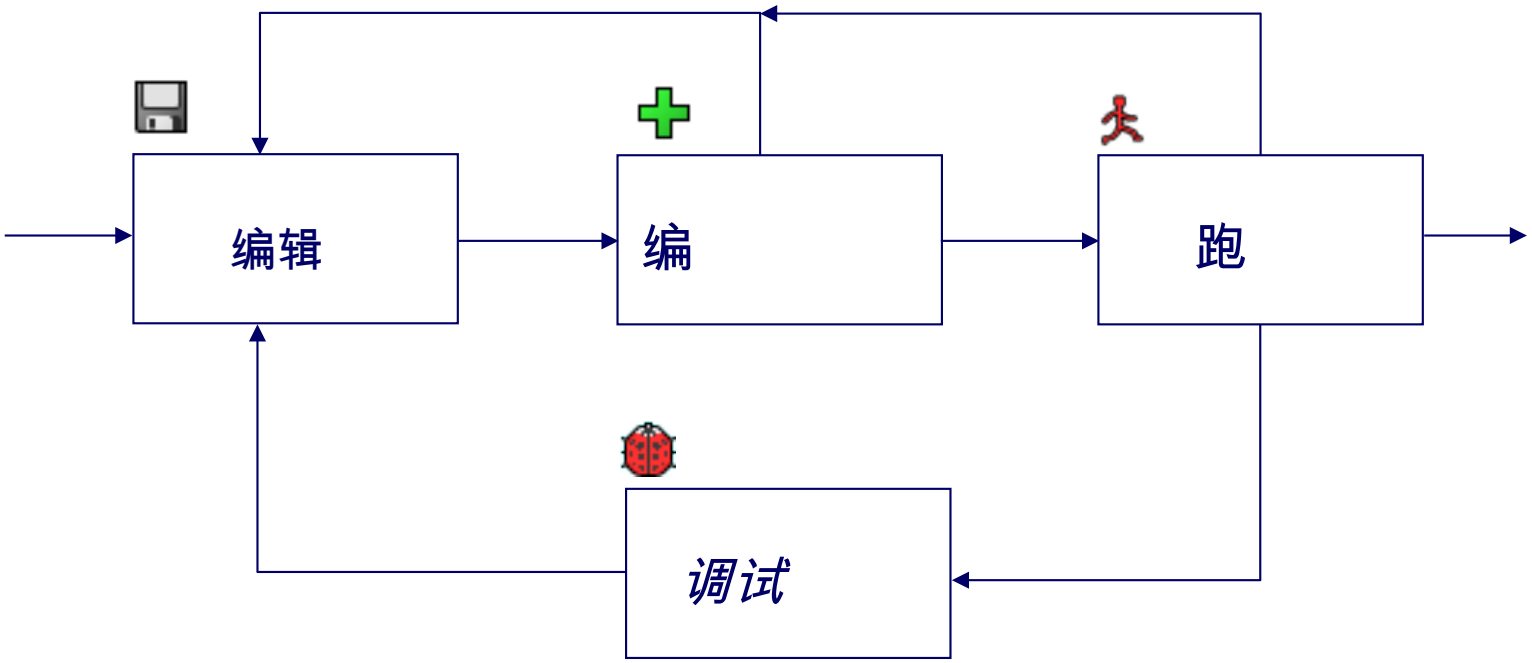
§ *Checkstyle的* 使用具有jGRASP支持编码标准，我们将在此过程中使用。

计划翻译

- **编译器 诉 解释（ Java虚拟机 ）**
- Java的编译过程：



实施周期...



- 这意味着周期增量程序建设。
- 计划尽早且经常重复这个循环。

语法和语义

- 语法：“语法”
  - § 如何词汇的规则可以用来组成
    法律结构的语言。
  - § 在程序方面，语言的语法
    介绍了如何形成法律语句和语言等结构。

- 语义：“意”
  - § 什么特定的法律结构，语言手段。
  - § 在节目中，语言语义上下文
    描述执行在语言中的合法语句时会发生什么。

语法和语义（续）

- 在自然语言中，有些东西是可以语法正确的，但没有任何意义.....

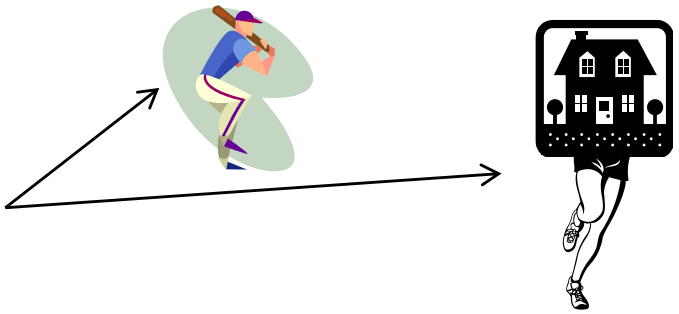
  § 蓝想法愤怒地睡觉。

- .....或在语法上是正确的，但有很多（可能的）的含义。

  § 时光飞逝像一个箭头。

  § 房子飞就像一个飞碟。

  § 你有没有看到一个本垒打？



- 编程语言不允许这些情况 - - 没有歧义！

  § 程序每次都会有相同的行为，它
  运行 - - 假设输入，如果有的话，是一样的。

程序错误

Q4 Q5 Q6 ___

- **编译时错误**
  - § 编译无法完成
    - 语法错误
    - 静态语义错误
  - § Java编译器不会产生字节码。
- **逻辑错误（逻辑错误）**
  - § 执行前进和正常停机，但不正确的
    行为或不正确的结果观察。
- **运行时错误**
  - § 执行异常中止。
    - 深结，碰撞，爆炸，玉石俱焚，大清洗
  - § 违规操作，异常。
- **查找错误 测试 并删除它们 调试**

编程语言概述

- 一种编程语言是一种人工语言设计给人以表达程序和有这些程序翻译成机器可执行的形式。

- 编程语言可以以不同的方式进行分类，例如：

  § 机器语言

  § 汇编语言

  § **高级语言（例如，Java，C ++，Python的）**

- 在不同类别的语言显然将是非常不同的对方，但在同一类别中，甚至语言都可以有很大的不同。

相同的程序，不同的语言

Java的

```
/ **从平原* / public类War_Eagle打印报价

{公共静态无效的主要（字串[] args）

    {的System.out.println（“沃尔伊格尔\ n！”）; }}
```

C / *打印报价从平原* /

```
主要（）
  {printf的（“沃尔伊格尔\ n！”）; }
```

阿达

```
- -  打印从Ada.Text_IO平原报价; 使用Ada.Text_IO; 程序War_Eagle是开始

     把（“战争鹰！”）; 新队; 结束Wa
  r_Eagle;
```

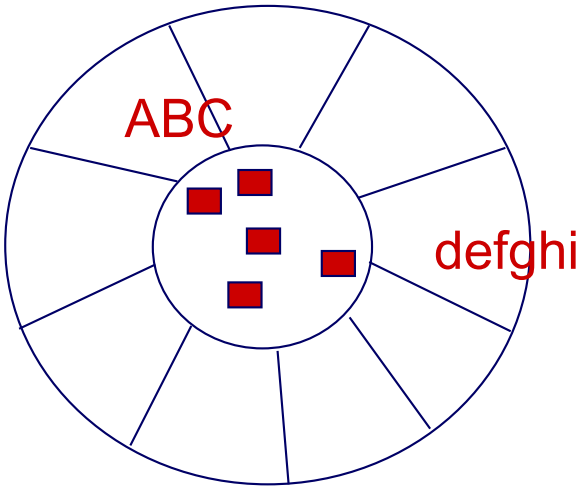
Perl的

```
#  打印从平原印刷报价，“\ n““战争鹰！”;
```

面向对象编程

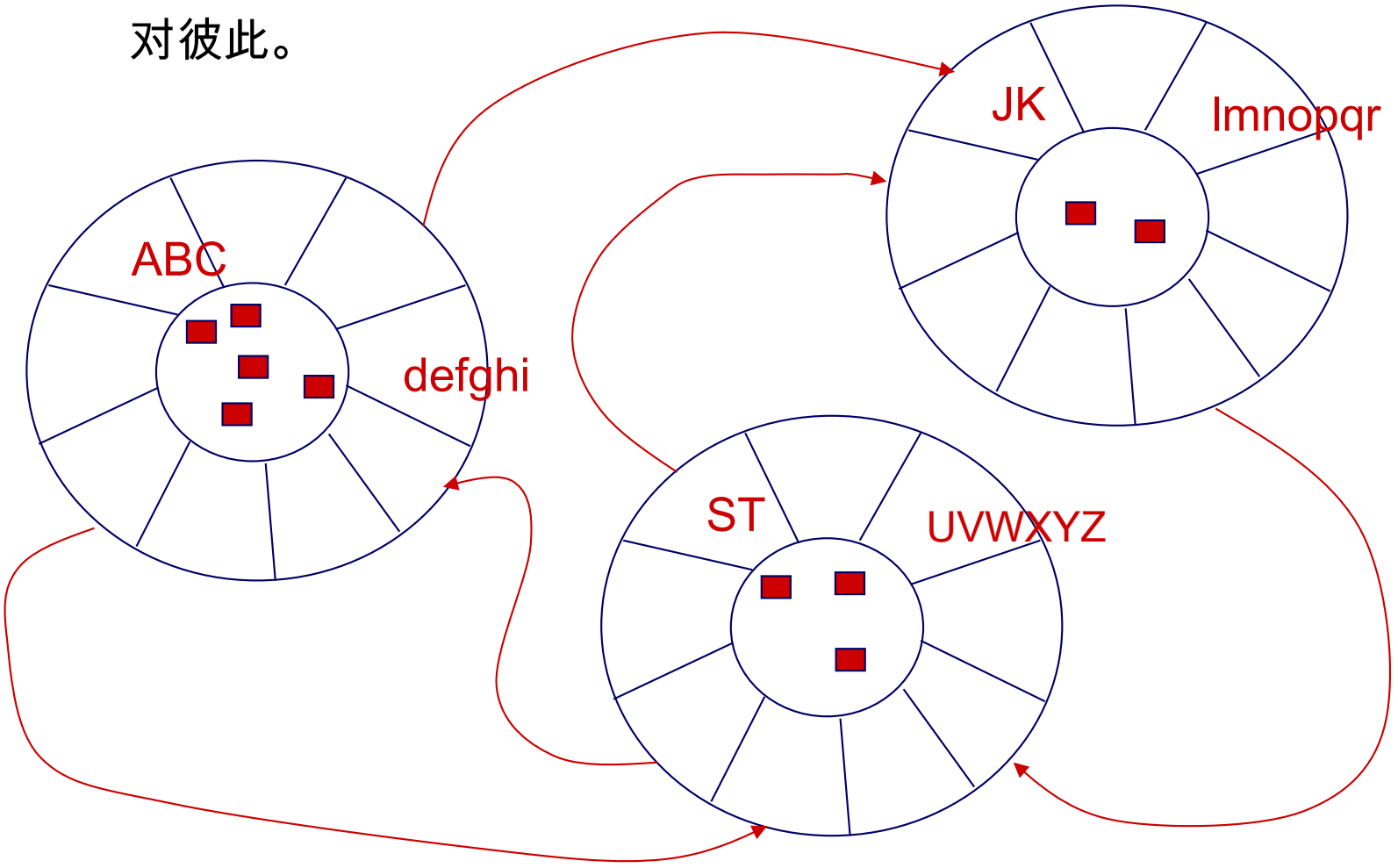
- OOP是一种编程的世界观中，在现实世界中的事物建模软件 **对象**。

§ 一个目的是实际上只是一个 **抽象化** 一个现实的
世界上的事情，作为一个实施 **封装** 私人 **数据** 和 **方法** （ 对数据的操作）。



面向对象编程（续）

- 对象通过发送沟通 **消息**
  对彼此。



面向对象编程（续）

- **CLASS =** 对象的一整类或组的描述

§ 通过现实世界的事物的类模型类

描述他们的“数据”和他们的“业务”。

类名称：数据的游戏玩家：等级

速度

健康加分

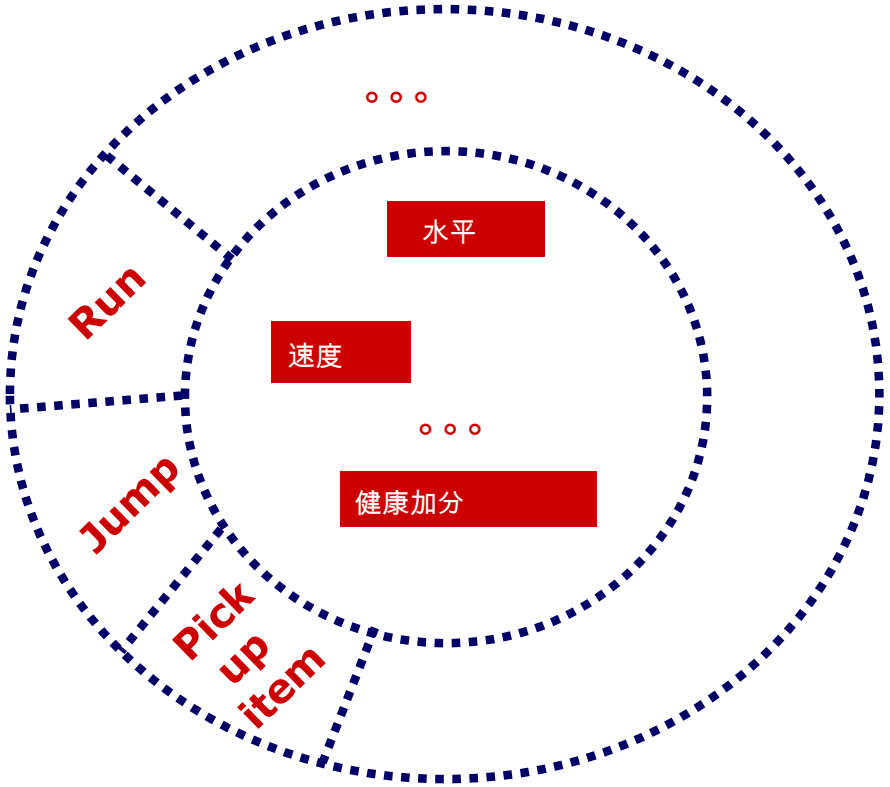
。。。

操作：

运行跳跃

拿起项目

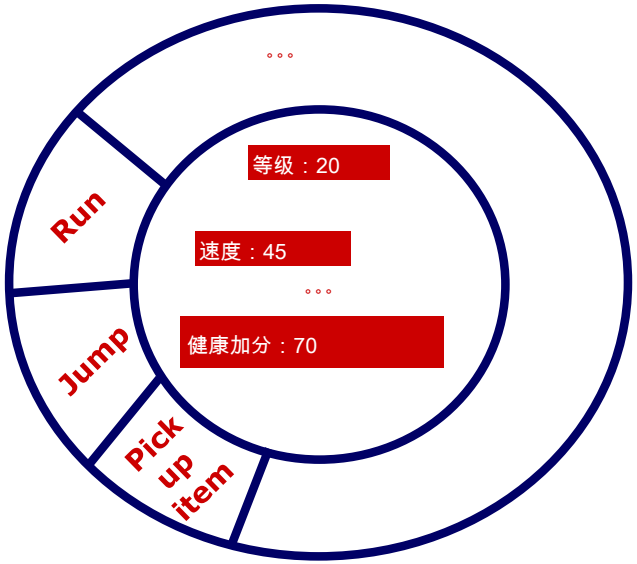
。。。



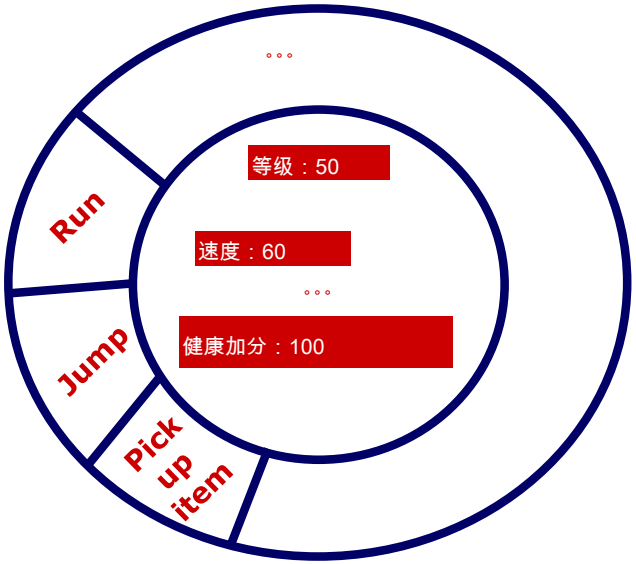
面向对象编程（续）

- 的目标是一 **例** 一些特定的类。

Player2



PLAYER1

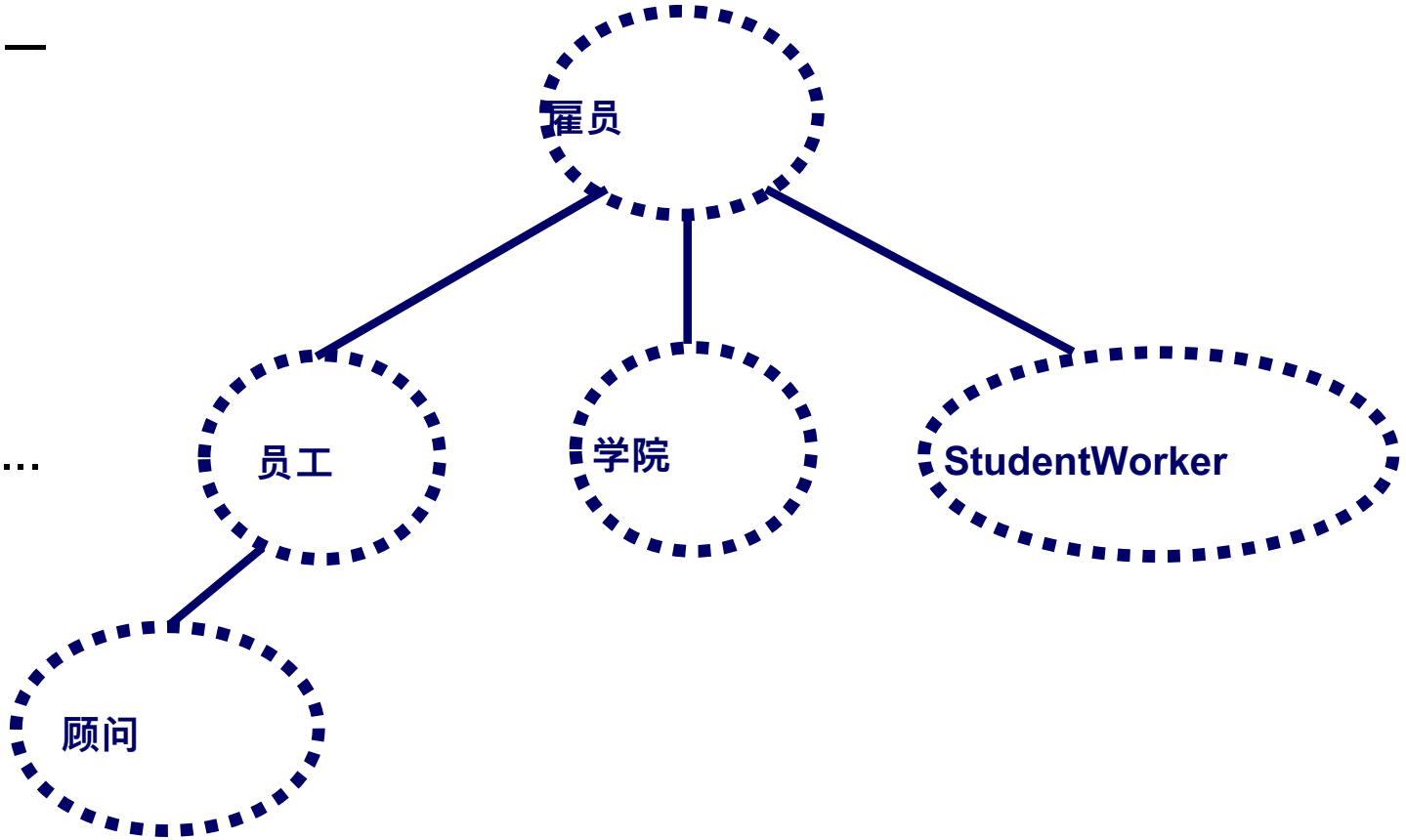


面向对象编程（续）

- 新类是从通过现有的衍生 **遗产**。

假设你正在创建一个程序来追踪

员工在奥本大学...



面向对象编程（续）

- OOP的目的是支持软件 **重用**。
- **类库** 此支持的一个重要因素。

§ 类库集的类设计为
可重用的组件，其服务可以被许多程序使用。

- Java应用程序编程接口（ **API** ）是自带的一组类库 **JDK**。

§ Java的API分为 **包** 如
java.awt中， java.io， java.lang中， 和 java.net

§ 例如：您在使用系统类的
输出语句是在java.lang包