

Stage1 学习报告

1. Vi

第一次在 Linux 下面使用 Vi 这个强大的文本编辑器,一开始使用很不习惯,和传统的编辑器模式很不一样.经过阅读教程和动手操作,我开始了解到 Vi 这个编辑器的独特之处.

Vi 总的来说分为三个模式,命令行模式,插入模式以及底行模式.每个模式下都有属于自己操作的命令.这些命令虽然有很多,乍看之下很难记住,但是通过操练和配置 sonar 的时候对 sonar 的环境配置等实际演练后,对几个常见的命令已经很熟悉了,但遇到记不起来的命令,回去翻阅一下文档,就能运用起来,逐渐不再对 Vi 编辑器产生排斥的感觉,相反觉得 Vi 的设计很有特色.

据说真正会用 Vi 的人可以将编辑文档敲代码的速度提升很多,我只是入了 Vi 的门,想要真正精通 Vi 并且将它的真正价值发挥出来,还需要以后在实际操作中一次又一次地练习.

总的来说,原生的 Vi 刷新了我对编辑器的认识,以前我觉得在命令行里面的操作都很奇怪很原始难以理解,但是学了 Vi 的教程之后,就像学了命令行操作 git 以及熟悉了 Linux 下命令行对文件的操作后,觉得这才是真正强大以及高效的手段.

2.Java

Java 对于有 C++基础的人来说,并不会很难上手,因为暑假的时候有自己学过一点安卓开发,所以算是对 Java 有一点点了解.

Java 有很多属于自己的特色,在 C++的基础上做了很多优化.比如删除了指针的概念,引入了安全指针----引用;引入了自动内存管理,让我们在编程的时候可以把精力从 new 和 delete 中抽离出来;函数传参传的是引用而不是值,降低了传参的代价;Java 中的 String 的值没办法二次赋值等等.

总的来说,除了课内学过的 C 和 C++之外,虽然 python,php,js 等都有接触过但是并不熟悉,Java 算是课外的我稍微熟悉一点的编程语言,尤其在开发安卓的时候,感觉到了 Java 的强大.

这次实训中更是学到了很有价值的 Ant,Sonar,Junit 等工具,想必在以后对 Java 编程的工作中会起到如虎添翼的作用.

3.Ant

这是一个有点类似 C++的 makefile 的工具,在认识它之前,对 java 文件的操作只能手动来一个一个做,直到 Ant 自动化改变了这一切,让编程工作一下子简单了不少!

Ant 的一大亮点,就是使用 Xml 格式来编写.对于有 web 前端基础,或者安卓,微软应用开发基础的人来说,Xml 并不陌生,甚至它是我们遇过的最好理解的语言之一,也正是因为这样,Ant 的上手相当快.

Ant 中第二个亮点,就是依赖关系.这有点像 makefile 里面的依赖,也有点像 Java 编程中 javac 命令会自动编译这个文件所引用的其他 java 文件一样,这种依赖关系保证了这个步骤的前提成立,确保这个步

骤可以顺利进行,同时省去了手动一步一步操作的繁琐性.

Ant 的第三个亮点,就是对文件的操作.单纯对文件的操作并不值得注意,但是 Ant 对文件的操作非常之智能化,兼容各种错误,简单地举个例子,加入 build.xml 中指明要删除一个文件,可是这个文件并不存在,ant 仍然不会报错而是当做这个文件已经删除成功来对待.这样看来,Ant 真是一个聪明的构建工具.

学到这个工具真是太幸运了,以后会在 java 编程中好好利用这个神器.

4.Junit

自从暑假接触 Java 之后,一直有听说单元测试这个东西,可是一直不知道是什么.直到实训的时候看到 Junit 的介绍和教程,才对单元测试这个东西有了基础的了解,并感受到了它对测试程序的重要性.

Junit 我选择在 Eclipse 中使用它,十分方便,导入 Junit 包在项目中后,对某个类进行单元测试的时候,Eclipse 会自动帮你生成一个已经完成百分之八十的测试文件,你只需要将你所要测试的语句写进去,就可以进行单元测试.

Junit 不仅仅是一个功能单一的单元测试工具,它考虑了许多测试的情境.比如,对多参数分情况的测试的支持,对死循环或者有时间要求的测试的支持,对抛出异常是否符合预期的测试的支持等等,这些具体化的情境使得 Junit 更加得实用和强大.

要在 Java 编程中学会对 Junit 的使用,来对我的程序进行单元测试,这个过程可能很繁琐,但是 Junit 已经帮忙简化了很多,通过单元测试,来让你的代码真正变得强壮和可靠!