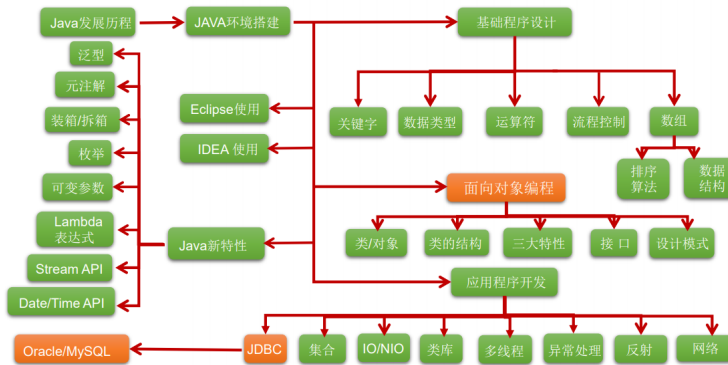


JAVA基础学习笔记

JAVA基础编程

java语言概述



第一部分：编程语言核心结构

主要知识点：变量、基本语法、分支、循环、数组、...

第二部分：Java面向对象的核心逻辑

主要知识点：OOP、封装、继承、多态、接口、...

第三部分：开发Java SE高级应用程序

主要知识点：异常、集合、I/O、多线程、反射机制、网络编程、.....

第四部分：实训项目

项目一：家庭收支记账软件

项目二：客户信息管理软件

项目三：开发团队人员调度软件

附加项目一：银行业务管理软件

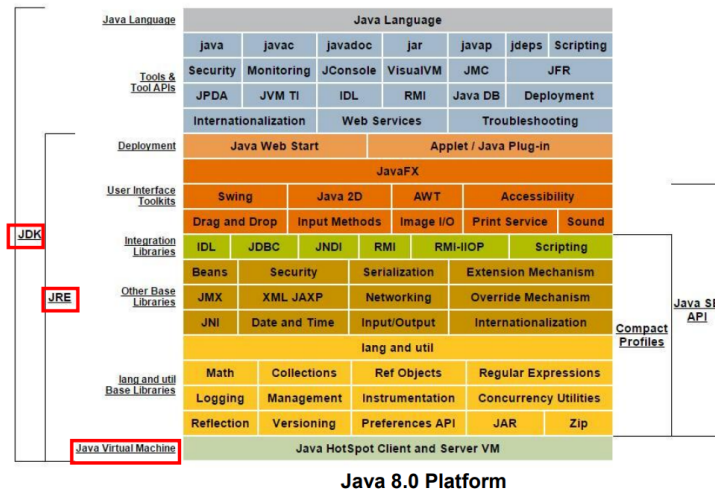
附件项目二：单机考试管理软件

基础知识

- 软件定义，分类（系统软件、应用软件）
- 人机交互方式：图形化界面（GUI）、命令行（CLI）
- $\text{programs} = \text{algorithms} + \text{data structures}$
- 计算机语言的发展迭代史
 - 机器语言 > 汇编语言 > 高级语言
 - 面向过程：C, Pascal, Fortran 面向对象：Java, Python, Scala

- Java语言版本迭代概述
 - 1996年，发布JDK1.0
 - 2004，发布JDK1.5，并更名为JDK5.0
 - 2014，发布JDK8.0，是5.0以来变化最大的版本
- Java语言应用
 - Java Web开发：后台开发
 - 大数据开发
 - Android应用程序开发：客户端开发
- Java语言的特点
 - 面向对象性
 - 两个要素
 - 类
 - 对象
 - 三个特征
 - 封装
 - 继承
 - 多态
 - 健壮性
 - 去除了C语言中的指针
 - 自动垃圾回收机制（仍会出现内存溢出、泄露问题）
 - 跨平台性

- write once, run anywhere, 基于JVM
- 开发环境搭建
 - JDK = JRE + 开发工具集 JRE = JVM + Java标准类库



- JDK的下载安装
 - 下载
 - Oracle官网
 - Github
 - 安装
 - 不能使用中文路径
- path环境变量配置
 - 为什么配置?
 - path环境变量是windows操作系统执行命令时会搜索的路径，需要java的开发工具（`java.exe`, `javac.exe`, `javadoc.exe`）在任何的文件路径下都可以运行成功
- 第一个JAVA程序



编写

创建Java源文件

```
class HelloWorld{  
    public static void main(String[ ] args){  
        System.out.println("Hello, World!");  
    }  
}
```

编译

javac HelloWorld.java

运行

java HelloWorld

troubleshooting

```
D:\>javac Test1.java  
javac: 找不到文件: Test1.java  
用法: javac <options> <source files>  
-help 用于列出可能的选项
```

- 源文件名不存在或者写错
- 当前路径错误
- 后缀名隐藏问题

```
D:\>java Test1  
错误: 找不到或无法加载主类 Test1
```

- 类文件名写错，尤其文件名与类名不一致时，要小心
- 类文件不在当前路径下，或者不在classpath指定路径下

```
D:\>javac Test.java
Test.java:1: 错误: 类Test1是公共的, 应在名为 Test1.java 的文件中声明
public class Test1{
      ^
1 个错误
```

➤ 声明为public的类应与文件名一致, 否则编译失败

```
D:\>javac Test.java
Test.java:3: 错误: 需要';'
    System.out.println("hello")
                        ^
1 个错误
```

➤ 编译失败, 注意错误出现的行数, 再到源代码中指定位置改错

• 注释与api文档

- 单行注释: //
- 多行注释: /* */
- 文档注释: /** */
- 作用: ①解释说明代码作用, 增加可读性; ②调试代码
- 特点: 单行和多行注释不参与编译, 文档注释可以被javadoc解析, 生成一套html格式的说明文档。
- Java API文档: API是语言提供的类库, API文档是提供的类库的说明书
- 良好的编程风格

- 正确的注释和注释风格
 - 使用文档注释来注释整个类或整个方法。
 - 如果注释方法中的某一个步骤, 使用单行或多行注释。
- 正确的缩进和空白
 - 使用一次tab操作, 实现缩进
 - 运算符两边习惯性各加一个空格。比如: `2 + 4 * 5`。
- 块的风格
 - Java API 源代码选择了行尾风格

```
public class Test {
    public static void main(String[] args){
        System.out.println("Block Style!");
    }
}
```

行尾风格

```
public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Block Style!");
    }
}
```

次行风格

- EditPlus的使用

- 开发工具

- 文本编辑工具:

- 记事本
 - UltraEdit
 - EditPlus
 - TextPad
 - NotePad

- Java集成开发环境 (IDE):

- JBuilder
 - NetBeans
 - Eclipse
 - MyEclipse
 - IntelliJ IDEA

- editplus下载与安装

- 777

- 77

- 77

- 77

以上内容整理于 [幕布文档](#)