# JAVA基础部分

1. jre(java runtime environment)java运行环境，包括运行的核心类库
2. jdk(java development kit)java开发工具包，jacac.exe(编译工具)，java.exe(运行工具)
3. 数据类型：整数（byte\short\int\long）浮点数（float\double）字符（char）布尔（boolen）类（class）接口（interface）数组（[]）
4. 变量：程序运行过程中，其值可以发生变化的量。
5. 标识符:数字、字母、下划线、美元符号$,区分大小写，不是关键字，不数字开头。
6. 命名规则：小驼峰(方法、变量)一个单词时小写，多个单词时第一个小写，其余首字母大写。
7. 命名规则：大驼峰(类)一个单词首字母大写，多个单词首字母都大写。
8. If语句：if(关系表达式){语句}/if(关系表达式){语句}else{语句}
9. Switch：switch(){case值1:语句1;break;case值2: 语句2;break;……default;语句体;break;}
10. Random:用于产生一个随机数。导包-创建对象-获取随机数
11. IDEA项目结构：项目/模块/包/类
12. 内容辅助语句：快速生成main()方法:psvm,输出:sout,内容提示/补全：ctrl+alt+space
13. 单行注释：ctrl+/；多行注释；ctrl+shit+t 格式化：ctrl+alt+L

# 前端部分

1. vue快速上手
2. 安装node.js;
3. Node.js 是运行在服务端的 JavaScript，
4. 如果我们使用 PHP 来编写后端的代码时，需要 Apache 或者 Nginx 的 HTTP 服务器，并配上 mod\_php5 模块和 php-cgi。
5. 从这个角度看，整个"接收 HTTP 请求并提供 Web 页面"的需求就不需要 PHP 来处理。对 Node.js 来说，概念完全不一样了。使用 Node.js 时，我们不仅仅 在实现一个应用，同时还实现了整个 HTTP 服务器。
6. NPM：是随同NodeJS一起安装的包管理工具
7. 允许用户从NPM服务器下载别人编写的第三方包到本地使用。
8. 允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。
9. 允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用。
10. 查看版本 npm -v
11. Windows npm 升级：npm install npm -g
12. Npm 安装某模块：npm install 模块名
13. 本地/全局安装：npm install express # 本地安装 npm install express -g # 全局安装
14. 查看安装信息：npm list -g 查看某模块版本信息：npm list 模块名
15. 卸载模块：npm uninstall 模块名
16. 卸载后，你可以到 /node\_modules/ 目录下查看：npm ls
17. 更新模块：npm 模块名
18. 搜索模块：npm search 模块名
19. PRPL(交互式解释器)类似shell 很好的调试 Javascript 代码
20. Node.js 异步编程的直接体现就是回调。
21. 异步编程依托于回调来实现，但不能说使用了回调后程序就异步化了。
22. Vue创建项目

* >npm create vue@latest
* > cd <your-project-name>
* > npm install
* > npm run dev

# 设计模式

1. UNL：unified modeling language 统一建模语言，用于描绘软件的标准蓝图。
2. Uml结构：视图、图、模型元素、通用机制。
3. 图：用例图、类图、对象图、包图、组合结构图、状态图、活动图、顺序图、通信图、定时图、交互概览图、组件图、部署图
4. Uml特点：工程化、规范化、可视化、系统化、文档化、智能化。
5. 类图：类封装了数据和行为，是面向对象的重要组成，类图用来描述之间的关系。
6. 类组成的三部分：类名、属性、类的操作。
7. 类之间关系：关联关系（双向、单向、自、多重、聚合、组合）、依赖、泛化、接口与实现。
8. 顺序图：表示对象之间的动态交互，即强调对象之间消息传递次序的交互图。
9. 执行者：小人（可没有）；生命线-----；对象：矩形（对象名称有下划线）；消息（对象之间的通信）：箭头；交互片段：alt（多路）、opt（任选）、par（并行）、loop（循环）；

# LINUX

1. 简介：多用户、多任务、多线程、多处理器、高稳定、高安全
2. 内核：进程调度、内存管理、虚拟文件系统、网络接口
3. 特点：模块化程度高、源码公开、设备独立性、广泛硬件支持、安全性可靠性好、可以执行强、用户界面良好、良好网络和文件系统支持
4. 用户名@计算机名 用户目录
5. 创建用户：adduser zhangsan
6. 修改密码：passwd zhangsan
7. 显示当前工作目录：pwd
8. 更改工作目录：cd
9. 创建工作目录：mkdir
10. 切换用户：su
11. 删除空工作目录：rmdir
12. 列出所有文件和目录：ls 包括隐藏：-a
13. 强制删除：rm -rf xxx 谨慎使用
14. 移动目录或文件：mv xxx xxx
15. 复制/重命名目录或文件：cp xxx xxx
16. 创建文件：touch xxx
17. 查看文件最后编辑时间：ls -l xxx
18. 查看文件最后访问时间：ls -lu xxx
19. 修改文件最后访问时间：touch -at 02011011 xxx
20. 创建和显示文件：cat xxx
21. 更改文件权限：chmod 777(组合权值) xxx
22. 分页显示：more/less xxx
23. 显示文件前n行：head n xxx
24. 显示文件后n 行：tail n xxx
25. 文件内容排序：sort xxx
26. 检查文件重复内容：uniq xxx
27. 文件中查找内容：grep ‘a’ xxx
28. 文件字节比对：cpm xxx xxx
29. 有序文件比对：comm xxx xxx
30. 文件内容查找：find -name “ex\*” （ex开头）
31. Fiand -atime -1 （1天内存取过）
32. 查看指定文件所在位置：where is ls
33. 查看可执行文件位置：which pwd
34. 显示磁盘空间：df -h
35. 列文件系统类型：df -T
36. 统计目录或磁盘占用空间大小：du -h xxx
37. 压缩文件：zip name name.zip;bzip2 name ;
38. 删除压缩包中文件：zip -d name.zip xxx
39. 压缩包中添文件：zip -m name.zip xxx
40. 覆盖解压缩：unzip -o name.zip; bzip2/bunzip2 name.bz2
41. 另存为解压缩：unzip -n name.zip
42. List压缩包信息：zipinfo name.zip
43. 用man查看ls命令帮助：man ls；help ls；info ls
44. 查看命令功能：whatis ls
45. 清除屏幕：clear
46. 显示文本：echo
47. 查看时间/日期：date ；cal；cal y；
48. 查看所有进程：ps -A
49. 查看某进程：ps -ef | grep xxx
50. 终止某进程：pkill xxx
51. 显示最近登陆系统的用户信息：last
52. 显示历史指令10条：history 10
53. 定义别名：alias l = ‘ls’
54. 删除别名：unalias l
55. Vi编辑器：命令模式、插入模式、底线模式
56. 按：进入底线模式；按a、i、o进入插入模式；esc退出；
57. 退出/强制修改退出：wq/wq！
58. 查看 more name
59. 保存内容：:w
60. 打开下/上一个文件：:n/N
61. 不存盘退出/强制不存盘退出：q/q！

# 版本控制系统

1. 版本控制记录文件内容变化，以便查看和修改对应的版本的系统，
2. 集中式版本控制工具（c/s模式）：svn、cvs、vss，优点：权限控制、会单点故障（服务器坏…）
3. 分布式版本控制工具：客户端本地有完整项目仓库，git、mercurial、bazaar、darcs
4. Git发展历史：1991年linus手动合并-2002年bitkeeper授权社区使用-2005年linus用c开发git-两周后linux由git托管-2008linus上线托管jquery、ruby、php……
5. 工作区：存放代码的位置。修改的内容没有历史记录。
6. 暂存区：工作区的代码需要添加到暂存区零时存储。内容可被销毁。
7. 本地库：提交到本地库的内容有了历史记录，内容不可销毁。
8. 代码托管中心（远程库）：局域网：gitlab、互联网：github、gitegitrj
9. 设置签名：git config --global user.name zhangyouze
10. 设置签名邮箱：git config --global user.email zyz18760877208@outlook.com
11. 签名信息存放在：C:\Users\zyz18\ .gitconfig
12. 签名的作用是区分不同的操作者，用户签名信息在每一个版本提交信息中可以看到，确认是谁的操作，git首次使用必须设置用户签名，否则无法提交代码。和远程仓库账号无关系。
13. 初始化本地库：获取该文件的git管理权限，初始化管理信息：在该目录下使用git init
14. 查看本地库状态：git status //on branch mast 当前项目在mast分支 no commits yet当前没有内容需要提交。
15. 使用linux命令创建文件并编辑：vim hello.txt \ i(插入模式)hello(插入内容) \esc \:(命令模式)yy(复制)\p(粘贴)\:(命令模式)wq（保存退出）\ll/l查看\cat hello.txt(打开)
16. **提交**到暂存区：git add hello.txt
17. **查看**：git status
18. **删除**暂存区文件：git rm --cached hello.txt（只是删除暂存区，工作区还是存在的）
19. **提交到本地仓库：**git commit -m “备注/版本信息” 文件名：git commit -m "first\_commit" hello.txt
20. 查看版本信息：git reflog /git log
21. 修改文件并查看：提示modifide
22. 版本回退 ：git reset --hard 版本号
23. 分支：在版本控制过程中，同时推进多个任务，我们可以为每个分支单独创建单独的分支，自己开发的时候不影响主干运行。
24. 查看分支：git branch -v
25. 创建分支：git branch hotfix(分支名字)
26. 切换分支：checkout hotfix
27. 把其他分支合并到当前分支：git merge hotfix
28. 冲突产生原因：当两个项目在同一文件，同一位置有两个不同的版本时，git无法替我们决定使用哪一个，必须人为决定内容。
29. 合并冲突：git commit -m “merge\_text”(不加名字)
30. 把代码推送到github：创建仓库/记下仓库http链接
31. 创建别名：git remote add note <https://github.com/Youzez/note.git>
32. 查看别名：git remote -v
33. 推送项目到分支：推送最小单位是分支，没有别名可用链接，