<!--

- @Description: 复习
- @Author: FallCicada
- @Date: 2024-12-05 16:48:46
- @LastEditors: FallCicada
- @LastEditTime: 2024-12-10 20:17:01
- @Slogan: 無限進步
  -->

Java basics

封装

- 是面向对象语言的三大特征，另外两个是继承和多态
- 封装 隐藏对象的属性和实现的细节 仅对外提供公共的访问方式
  - 提高安全性
  - 用户不需要知道内部具体复杂的实现。只需要使用对应的功能即可
- 封装的原则
  - 把不需要对外提供的内容隐藏起来
  - 把属性隐藏掉，提供方法来修改，保证数据的安全
    - 属性：

```
  余额
stu1.name = "张三";
card.balance = 10000;
```

- 封装的实现
  1. 用 private 修饰成员属性（成员变量）
  2. 提供对应的公开的 public set和 public get方法

继承

- 继承是属于类与类之间的一种关系
- 继承是面向对象语言的三大特征之一
- java中类继承是单继承，一个子类只能继承一个父类(一个父亲)
- java中不能多继承，以下代码编译报错

```
public class Student extends Person,Animal
```

- 多层继承
  - 子类A继承父类B，
  - 父类可以继承爷爷类C
  - A extends B{}
  - B extends C{}

多态

- 面向对象语言的三大特征之一
- 一种事物，有多种状态
- java中多态怎么实现
    - 子类继承父类
    - 子类要重写父类的方法
    - 父类的引用指向子类对象（一个父类的引用可以指向任何一个子类的对象）

```
Animal a = new Dog();
```

    - 父类的引用调用子类重写的方法
    - java中完全相同的代码出现在不同的位置，执行的结果不同
        - point.show();

编写九九乘法表

```
public class Main {
    public static void main(String[] args) {
        for (int i = 1; i <= 9; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(j + "*" + i + "=" + (i * j) + "\t");
                if (j == i) {
                    System.out.println();
                }

            }
        }
    }
}
```

Java Advanced

比较器

I/O流

MySQL

生命周期

- 开始

  MySQL会给事务分配一个唯一的id，并且开始记录事务日志

- 执行

  可以执行多个sql语句，包括insert、update、delete等操作，会在事务日志中进行记录，并不会影响数据库的实际数据

- 提交

  所有sql语句执行完毕之后，没有发生错误，可以选择提交事务，事务中的所有修改，永久保存到数据库，操作生效

- 回滚

如果发生了错误，或者用户决定放弃事务，可以选择回滚事务，回滚会撤销事务中的所有修改，恢复到事务开始前的状态

- 结束

  事务的生命周期在提交或者回滚之后结束。结束后会释放相关资源，关闭事务日志。

ACID

- 原子性 `Atomicity`

  一个事务中的所有操作，要么全部完成，要么全部不完成，如果事务在执行过程中，有错误发生，会被回滚。

  1. 自己账户 -100

     停电了

  2. 对方账户 +100

- 一致性 `Consistency`

  事务执行前后，数据库必须处于一致状态，满足约束等

- 隔离性 `Isolation`

  并发环境不同的事务操作相同的数据时，每个事务都有各自完整的数据空间，并发事务所做的修改，必须与其他事务做的修改隔离。事务查看数据更新时，数据的状态要么是另一事物修改前的状态，要么是修改后的状态，事务不会看到中间状态的数据。

- 持久性 `Durability`

  只要事务被成功提交，对数据的操作就要永久的保存下来，即使系统崩溃，重启系统之后，数据库还能恢复事务成功结束时的状态。

  1. 自己账户 -100
  2. 对方账户 +100

  停电了

开始前索引

语句

```
## 启动数据库
## 1. 创建数据库
CREATE database +数据库名;

## 2. 查看数据库
show databases;

## 3. 选择数据库
use +数据库名;

## 4. 创建表
CREATE TABLE +表名()

## 5. 查看表的结构
desc +表名;
chow columus form +表名;

## 5. 添加字段
```

```
insert into +表名(+字段名) values(+值);

## 6. 删除表
drop table +表名;

## 7. 查看表内容
select * from +表名;

## 8. 左外连接
select * from 表名1
left join 表名2 on 表名1.id = 表名2.id;

## 9. 更新表
update +表名
set +字段名 = +值
where +条件;

##
```

front end

html文件怎么写 主要有哪些部分著称

```
<!DOCTYPE html> <!-- 声明文档类型为HTML5 -->
<html lang="en"> <!-- 定义HTML文档的语言为英语 -->
<head>
    <meta charset="UTF-8"> <!-- 设置文档的字符编码为UTF-8 -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!--
设置视口，以确保页面在移动设备上正确显示 -->
    <title>Document</title> <!-- 定义文档的标题，在浏览器标签栏显示 -->
</head>
<body>
    <!-- 页面内容将放置在这里 -->
</body>
</html>
```

定义网页标签

```
<!-- 视频标签-->
<video src="xxx" controls></video>

<!-- 音乐标签-->
<audio src="xxx" controls></audio>

<!-- 图片标签-->
<img src="xxx" alt="">
```

Git

- `git init`：初始化一个新的Git仓库。在当前目录下创建一个新的Git仓库。
- `git add`：将文件添加到暂存区（staging area）。可以指定文件名或使用通配符。

- `git commit -m`：提交暂存区的文件到本地仓库，并添加提交信息。-m 选项后跟提交信息。
- `git remote add`：添加一个远程仓库。通常用于将本地仓库连接到一个远程服务器。
- `git push`：将本地仓库的提交推送到远程仓库。通常用于将本地更改发布到远程服务器。
- `git clone xxx`：克隆一个远程仓库到本地。xxx 是远程仓库的URL。
- `git pull`：从远程仓库拉取最新的更改并合并到本地分支。相当于 git fetch 和 git merge 的组合。
- `git status`：显示当前工作目录的状态，包括已暂存的、更改的和未跟踪的文件。

项目配置与管理

- 选择题*20
- 填空题*10
- 判断题*10
- 简答题*5

maven

配置文件内容

命令

用途

生命周期

用途/作用

Spring

重要注解

配置文件内容

Vue

遍历元素

文件拓展名

官方路由管理工具’

表单，表单数据的使用

Linux

- 选择题*20
- 填空题*10
- 判断题*10
- 简答题*4

Vi编辑器

服务器装Java环境

Linux命令

- `ls`：列出当前目录下的文件和目录。
- `cd`：切换目录。
- `pwd`：显示当前工作目录。
- `mkdir`：创建一个目录。
- `rm`：删除文件或目录。
- `mv`：移动或重命名文件或目录。

文件操作

创建用户