面向对象答辩

项目实际案例-推荐算法

前言

该项目是上家公司面向对象的实际案例，公司从事招聘行业，其中一核心技术在于精确推荐职位。

需求

根据求职者填写信息与职位匹配程度高低按权重值大小实现排序。

## 算法

划分为多个维度（权重影响因素），例如职位、薪资、距离、发布时间等，各个维度权重不一致，算法不同。

Y = ax1 + bx2 + cx3 + dX4 +ex5+ fX6 +g；按Y值进行排序，Y值越大越靠前。

a = 薪资权重， b =距发布时间权重值， c =公司匹配度权重值，d =用户自己投递过得相似职位权重值， e人工调整其中a,b,c,d,e 可实现灵活配置。

x1 , x2 , x3, x4得分范围： 0-10分

x1=薪资；x2=发布时间；x3=公司；x4=用户自己投递过的职位；

以下为各维度得分规则：

x1 薪资：期望薪资中线上浮20%，做为目标中线，职位的薪资中线和目标中线对比得分，用衰减函数控制；获取不到薪资时，则为零分。

设期望薪资下限min，设期望薪资上线max；职位薪资下限为jdmin，职位薪资上限为jdmax

目标中线: O = (max+min)/2 \* 1.2

职位中线：J = (jdmax + jdmin)/2

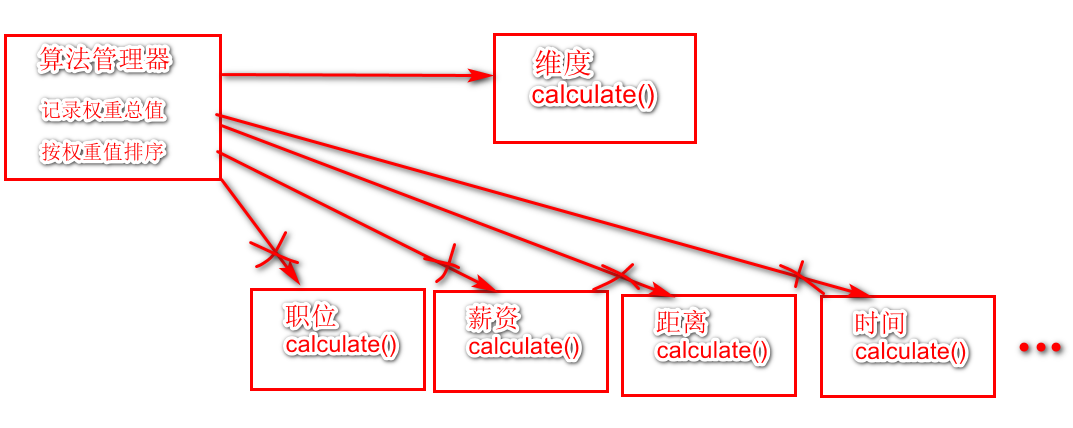
衰减速度：若max = min，S= max/10；否则S = (max-min)/4

**X1 = 10 - |J - O|/ S;** 若X1 < 0 ,则X1取值为0.

例如：期望薪资为10000～20000，中线为15000，上浮20%为18000，每隔2500衰减一分，则一个职位中线为18000的得分10分，职位中线为15500的得分9分，职位中线为20500的得分9分。

**X2 距发布时间：**x2 = 10 \* 0.9d  ，d为职位发布时间与当前时间的差值，天粒度。获取不到发布时间，则为零分。当d > 14时，X2取值为0。

推荐算法系统架构设计



# 三大特征

封装（分）

考虑业务的复杂与多变，利用封装“分而治之，变则疏之”的设计原理，化分为多个类，算法管理器、职位类、薪资类、距离类、上传时间类等。

继承（隔）

职位类、薪资类、距离类、时间类为需求变化点，根据业务需要逐步在优化，所以需要对职位类、薪资类、距离类、时间类进行抽象，去掉不同，抽离共性，定义维度类，即父类，算法管理器只需调用父类维度类，无需调用子类，达到隔离变化的目的。

多态（做）

提到多态首先考虑的是3大步骤：

1、调父，算法管理器调用父类维度类；

2、重写，子类职位类、薪资类、距离类、时间类重写父类维度类计算方法；

3、根据求职者填写信息调用职位类查询方法查看是否有相匹配信息，若有，创建子类对象，再调用子类计算方法。

目的：项目扩展性强，当有需求变化，只需修改子类方法，即隔离了算法管理器，也保留了子类计算方法的独立性，满足了开闭原则。

# 六大原则

开闭原则-重点在于“闭”，新增其它维度类，算法管理器不做改变

单一原则-各个类职责单一，只围绕一件事情来展开，算法管理器，根据权重大小实现排序；职位类，提供职位计算方法；薪资，提供薪资计算方法。

依赖倒置-客户端调用父，不调用子，算法管理器调用维度类，不调用职位类、薪资类、距离类和时间类。

迪米特-要求各个类之间低耦合，依赖性要低，具体体现职位类、薪资类、距离类、时间类低耦合，各个子类与算法管理器低耦合。

组合复用-复用代码优先考虑组合复用，客户端算法管理器通过调用父类维度类，传入子类对象来实现重写父类方法的目的。

里氏定律**-**2步骤，

1、创建对象时，调用父类地方传入子类对象(本项目未做体现)；

2、子类职位类、薪资类、距离类、时间类重写父类计算方法，各个子类根据自身算法进行扩展。

附加：面向对象典型案例，个人简历功能

需求：简历的基本数据有ABCDEF项，根据业务需要，要求做行业细分，区分白领简历和蓝领简历，那么就创建白领简历这个对象，它继承“简历”对象，自动就有了简历基本的结构，然后在这个基础上再写白领建立需要扩展出来的部分，这样就节约了开发时间，同时如果简历基本数据变更，只要修改“简历”对象，“白领简历”自然就有了简历对象的基本方法。