

实验二 软件需求的优先级排序

实验目标：

- 为软件需求进行优先级排序

实验思路：

- 选定开源IDE项目：NetBeans
- 明确信息源：StackOverflow
- 需求获取：从StackOverflow中爬取以ide和netbeans为标签的数据
- 需求的优先级排序：对爬取的数据进行分析，按照频率，浏览量等进行排序
- 效果分析：根据eclipse上的缺陷报告，对需求优先级的划分进行验证

实验步骤：

目标IDE与信息源：

- NetBeans 最新下载量已经超过18万次，参与开发人员超过80万，用户量大，可以爬取的信息充足，可以较为准确地进行需求分析。
- StackOverflow 是一个IT技术问答网站，平台上面有很多有关ide以及NetBeans相关的问答，有很多专业人士提出的使用过程中的问题，能够较为真实的反应ide中存在的缺陷或未完善的需求。

需求获取：

- 文件：

- * `creeper.py` 用于爬取标签为IDE和NetBeans的相关问答
- * `creeprank.py` 用于爬取每条问答的相关的投票数/回复数/浏览量

- 获取思路：

通过爬虫从StackOverflow爬取相应的问答，并且爬取每条问答的点赞数、回复数以及浏览量，这些数据可以作为优先级的划分依据

数据文件：

- 文件：

- * `ide.txt` 以ide为标签的相关问答数据
- * `netbeans.txt` 以netbeans为标签的相关问答数据
- * `ide_rank.txt` 以ide为标签的相关问答数据的投票数、回复数以及浏览量
- * `netbeans_rank.txt` 以netbeans为标签的相关问答数据的投票数、回复数以及浏览量

数据分析：

- 文件：

- * `frequentAnalyse.py` 词频分析程序，用于对爬取的数据进行相应的词频计算以及排序
- * `summary.py` 将词频分析程序的结果跟投票数、回复数以及浏览量进行整合

- 分析思路：

先将以**IDE**为标签和以**NetBeans**为标签的数据进行词频分析，将所得的结果跟相应的投票数、回复数以及浏览量相对应，对数据进行整合。根据实验1的分类，对每种需求对应的相关单词进行处理，从整体的数据划分其优先级。

分析结果：

- 文件：

* **word_freq.txt** 使用词频分析程序处理后的数据结果，包含出现频率最高的60条数据，其格式为：
|单词|频率|
* **sortbase.txt** 根据词频汇总所有的数据，其格式为：|单词|频率|投票数|回复数|浏览量|

优先级划分

1. 对所有的数据进行了词频排序，以及投票数、回复数、浏览量的整合，所得的结果：

- 代码、文本编辑功能：2513 1480 2765996

单词	频率	投票数	回复数	浏览量
code	1825	6711	3021	5848951
file	2115	6264	4222	7371951
files	960	1966	1464	2221456
editor	365	1601	851	1413089
line	301	4339	1391	3492248
text	900	1126	988	1720786
change	205	1638	1005	1957902
duplicate	772	1726	1081	2500643
source	182	1609	1124	2005053
color	106	642	436	695871
find	458	1477	968	1935493
data	386	1064	1216	2028510

- 代码分析、编译、调试功能：1812 1281 2145446

```
* error      1822    2967    2075    4115476
* class      1024    2080    1809    3292610
* new        145     907     680     1353133
* build      673    1758    1288    2493351
* debug      130    2131    1230    2152119
* debugging  130     930     357     658372
* run        967    3022    2212    4428711
* running    409     704     601     829800
```

- 多语言编辑：491 334 621389

* python	358	466	333	460466
* php	959	1146	658	1141055
* javascript	192	248	176	330583
* html	129	336	260	381126
* mysql	441	263	243	793718

- 多环境编辑: 543 456 701131

* android	465	610	312	728340
* windows	579	301	350	327820
* linux	149	118	163	128096
* web	789	1145	999	1620271

- 文件管理: 3161 1494 3258025

* closed	1288	8502	3187	7724952
* open	197	1409	1013	1904150
* add	722	2045	1320	2580086
* library	336	688	457	822913

- IDE平台: 182 72 110671

* phpstorm	278	447	51	200971
* pycharm	219	104	43	50922
* delphi	211	59	50	28275
* vs	181	390	265	333007
* xcode	154	86	20	42202
* webstorm	135	11	5	8649

- 功能扩展: 1506 753 1442351

* development	259	1332	687	1171558
* auto	172	2623	1181	2223136
* find	160	1477	968	1935493
* plugin	470	1430	778	1475791
* selenium	139	36	40	68109
* support	137	1588	674	1432400
* method	488	2058	947	1789970

- 项目管理工具: 433 417 751168

* jar	838	898	896	1412148
* maven	713	504	402	966612
* glassfish	486	216	195	353712
* tomcat	371	115	177	272202

- 性能需求: 1047 781 1652779

* server	562	1047	781	1652779
----------	-----	------	-----	---------

- 功能需求: 469 417 811773

* application	894	1163	1200	1978583
* database	670	705	755	1569940
* gui	483	535	159	706142
* jframe	406	95	103	100817
* javafx	345	180	114	218770
* swing	340	139	181	296389

2. 优先级排序：

- 我们将得到的需求分为五级：Highest, High, Medium, Low, Lowest
- 根据上述数据我们可以进行分析，计算平均值得到下表：

需求	投票平均数	回复平均数	浏览量平均数	平均数
代码、文本编辑功能	2513	1480	2765996	923329
代码分析、编译、调试功能	1812	1281	2145446	716179
多语言编辑	491	334	621389	207404
多环境编辑	543	456	701131	234043
文件管理	3161	1494	3258025	1087560
IDE平台	182	72	110671	36975
功能扩展	1506	753	1442351	481536
项目管理工具	433	417	751168	250672
性能需求	1047	781	1652779	551535
功能需求	469	417	811773	270886

文件管理 > 代码、文本编辑功能 > 代码分析、编译、调试功能 > 性能需求 > 功能扩展 > 功能需求
> 项目管理工具 > 多环境编辑 > 多语言编辑 > IDE平台

优先级验证：

- 对于我们之前所做的优先级排序，进行相应的验证。
- 我们在eclipse上的缺陷报告网页查询有关关键词，对于IDE需求来说，我们认为它与优先级跟严重程度有关，因此我们整理了相关关键词所出现的bug所有优先级为P1或者严重程度为blocker或者critical的数量，整理成如下表格：

需求	平均优先级
代码、文本编辑功能	2513
代码分析、编译、调试功能	1812
多语言编辑	491
多环境编辑	36
文件管理	36
IDE平台	8
功能扩展	49
项目管理工具	50
性能需求	68
功能需求	52

- 从上述表格中可以看出，优先级的顺序应该为：

性能需求 > 功能需求 > 项目管理工具 > 功能扩展 > 代码分析/编译 > 文本编写 > 多环境=文件管理 > 多语言 > IDE平台

误差分析：

- 从上述得出的结论来看，我们做出的分析与验证的结果有一定的出入，进行误差分析。
- 首先，我们得出的结果中，文件的代码编写/编译运行/文件管理都处于比较高的优先级，但是在验证的时候，它们均处于一个中间水平的优先级，我们认为，这个实验误差是可以接受的，因为我们在需求分析的时候，代码的编写和编译是通用功能，出现的关键词较多，导致其他数据较高，因此高于预期值是在预想范围之内的。
- 除去上述三个需求，性能需求跟功能需求、功能扩展、项目管理工具虽然在细节方面的排序跟与有一定的出入，但是都跟验证的情况较为相近，都是优先级比较高的需求。
- 多环境编辑、多语言编辑以及IDE平台的需求都是处于较低的优先级，这个也是符合预期的。

得出结论：

- 综合上述情况，我们可以大致判断出：
Highest：性能需求
High：功能需求/功能扩展/项目管理工具
Medium：代码分析/编译/文本编写
Low：多环境/多语言编写
Lowest：IDE平台需求

成绩分配：

- 小组成员平均分配。