

# 程序设计在线作业代码不规范检测方法及应用

杜彬,贺杰,徐阳,王林鑫,尤枫  
(北京化工大学信息科学与技术学院,北京 100029)

**摘要:**针对程序初学者编写的C/C++程序,调试需要消耗大量的时间和精力。然而,初学者在接触简单的编程时,编写思路一般不存在问题,但由于其良好的编程习惯还未养成,代码格式往往不规范,从而导致程序相应代码块出错。本文结合程序在线评测系统,针对程序设计课程的C/C++程序代码不规范问题提出了一种基于正则表达式和编辑距离的检测方法并通过实验验证了不规范的代码出现错误概率较大。

**关键词:**软件测试;代码格式规范;程序在线评测系统

中图分类号:G623.58

文献标志码:A

文章编号:1674-9324(2018)44-0154-02

## 一、引言

目前国内外的程序自动化调试研究中,在针对特定类型故障、数据结构及内存等方面取得了一定进展,但对由不良程序编程习惯或者编程代码格式的不规范引起的错误类型研究还较少<sup>[1]</sup>。编程者往往在编程中只注重算法和正确性,却忽略代码规范性,消耗了软件调试过程中大量的时间和成本<sup>[2]</sup>。本文结合作者所在学校的在线编程评测系统(Online Judge System,OJ)<sup>[3,4]</sup>,分析系统数据库中答题者提交的程序,针对其中由于代码块括号匹配不规范出错的程序,进行错误程度计算,提出一种代码不规范程度检测方法。

## 二、代码格式规范化概念

就软件开发的代码编写而言,每个开发者的编程水平、经验和习惯都大不相同。本文对于程序代码格式规范化概念进行简单梳理,归纳如下:

1.格式一:对齐与缩进。(1)①“{”位置的两种风格:①“{”和“}”独占一行,且位于同一列,与引用他们的语句左对齐,便于检查配对情况;②“{”位于引用他语句之后,“}”与引用他的语句左对齐。(2)位于同一层“{”和“}”之内的代码相对于引用他们的代码缩进。(一般用4个空格的tab键缩进,不用空格缩进)。

2.格式二:代码行内空格。(1)关键字之后加空格,关键字与操作符之间不加空格。(2)采用第一种“{ ”格式的函数名之后不加空格,采用第二种“{ }”格式的函数名之后加空格。(3)赋值、算术、关系、逻辑等二元运算符前后各加一空格,但是一元运算符前后一般

不加空格。(4)“( )”中,变量之间紧跟“, ,”。(5)“, ,”;后留一个空格。

3.格式三:代码行。(1)一行只写一条语句。(2)一行只写一个变量。(3)每个关键字之间要加一个空格隔开。

## 三、代码格式不规范化程度检测方法

本文以OJ平台学生所提交的所有代码作为分析源,结合正则表达式及编辑距离的概念,使用Python对数据库中的代码计算相应的不规范程度。

1.正则表达式。本文设计的正则表达式如下。

(1)“.\*? +”用该模式去掉多余的空格。(2)[,! =<>\+|-|^\*V]+[A-Za-z0-9]+[!!=<>\+|-|^\*V] +”用该模式添加标识符与运算符之间的空格。

2.编辑距离。编辑距离用来度量两个字符串的相似度。基于编辑距离算法进行动态编程,其算法的时间复杂度为O(m\*n),空间复杂度为O(m\*n),m,n分别表示源字符串S和目标字符串T的长度。编辑距离的动态规划求解方程组如下所示。

$$D_{[i,j]} = \begin{cases} 0, & \text{if } i=0 \text{ or } j=0 \\ \min(d_{[i-1,j]} + 1, d_{[i,j-1]} + 1, d_{[i-1,j-1]}), & \text{if } S_i = T_j \\ \min(d_{[i-1,j]} + 1, d_{[i,j-1]} + 1, d_{[i-1,j-1]} + 1), & \text{if } S_i \neq T_j \end{cases}$$

## 四、实验验证

1.代码格式不规范化错误统计及实例。本文针对目前OJ平台数据库中提交的135,556份C/C++程序(包含提交结果正确和错误的代码),进行分析统计,计算全部代码的不规范化程度,结果如图1显示。横轴以规

收稿日期:2018-03-14

基金项目:本文由以下项目联合资助:北京化工大学研究生教育教学改革项目,“互联网+”形势下高校研究生安全管理微信平台体系,项目编号:G-JG-PT201602;北京化工大学本科生教育教学改革项目,“科研项目+竞赛”驱动的计算机专业人才创新实践能力培养与实践,项目编号:B201408

作者简介:杜彬(1971-),女(汉族),河北鹿泉人,副教授,博士,研究方向:计算机应用。

范代码及不规范代码之间的编辑距离作为划分依据，纵轴是属于此编辑距离的不规范代码数量。从图1我们可以发现，OJ平台数据库中，只有不到11,000份代

码做到规范化，当然如果把不规范程度在[1—20]之间的代码看作规范的话，那么也只有不到46,000份代码是规范的。

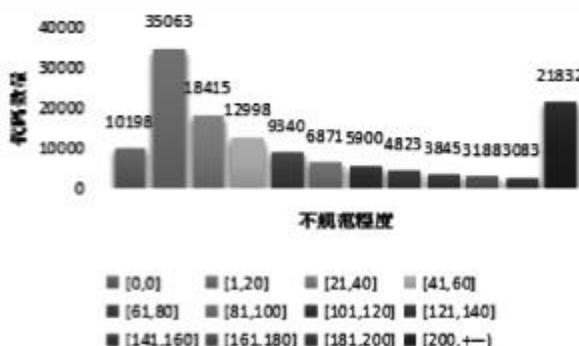


图1 全部代码的不规范程度统计

为了进一步分析不规范程度对于代码的正确性影响，我们将OJ平台数据库中87,933份错误代码进行

单独分析，图2显示了错误代码中不同规范化程度的代码分布情况及占总错误代码的百分比。

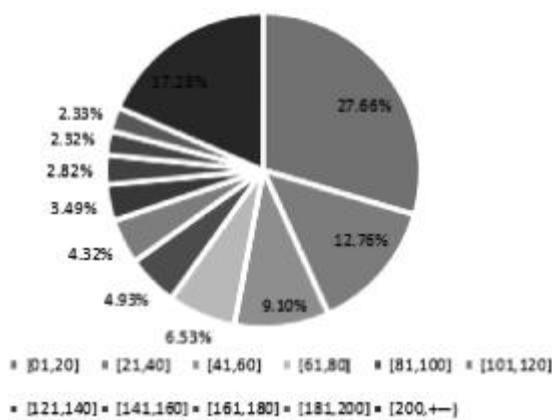


图2 错误代码中不同规范化程度的代码分布情况

我们综合比较图1和图2，可以观察到，错误代码中不规范代码的比例与总代码中不规范代码的比例成正相关。

## 五、总结与展望

本文针对C/C++程序代码不规范问题，提出了一种基于在线编程评测系统(OJ)的代码规范化检测方法框架，实验验证了该方法能够有效度量代码不规范错误程度，且代码不规范是程序错误的重要原因。之后的工作，我们将进一步分析程序员的编程习惯，并着眼于Java、C#以及Python等更多程序语言的代码不

规范自动修复问题。

## 参考文献：

- [1] 马春燕,刘杰,赖文豫.基于变异技术的程序故障自动化修复方法[J].计算机应用研究,2014,31(1):177-181.
- [2] 王延青,王建政,张丽杰,等.程序设计语言教学中编码标准的定量评测框架[J].合肥工业大学学报(社会科学版),2008,(6):67-71.
- [3] 李文新,郭炜.北京大学程序在线评测系统及其应用[J].吉林大学学报:信息科学版,2005,23(2):170-177.
- [4] 纪洪波.基于jQuery的Web源程序在线评测系统的设计与实现[D].吉林大学,2010.

## Standardization Method for Online Code of Programming Contest

DU Bin, HE Jie, XU Yang, WANG Lin-xin, YOU Feng

(College of Information Science & Technology, Beijing University of Chemical Technology, Beijing 100029, China)

**Abstract:** For inexperienced programmers, it costs lots of effort to debug and fix C/C++ program errors. When beginners get in touch with the simple programming problem, the solution methods are always correct. Due to their poor programming experiences and inadequate programming habit, the code formatting is incorrect, which leads to incorrect outputs. Motivated by the prospect of reducing human developer involvement, this paper proposed a detection method, especially for the code segment standardization, and applied this method to detect bugs of incorrect code formatting of C/C++ language programs.

**Key words:** software testing; code segment standardization; online judge system