

1

代码风格与代码风格规范化辅助工具

阿丽亚娜-5的故事



1996 年 6 月 4 日，欧洲航天局的阿丽亚娜-5运载火箭发射后 37 秒自行引爆。整个损失大约在 5 亿美元。

这样的事故是怎么发生的呢？

阿丽亚娜-5的故事

- 爆炸来自于未被处理的软件异常导致的自毁；
- 这个软件异常存在于不被需要的代码片段中；
- 异常来自于一个代表水平值的 64 位的浮点数值转成了 16 位有符号整数值，且这个值在 16 位整数表示下是会溢出的；
- 未被处理的这个异常导致了整个软件无法正常工作，从而导致了计算机系统无法正常工作，进而导致了发射任务的失败。

墨菲定律

- “凡是可能出错的事就一定会出错”（怕什么来什么）；
- “如果一个事情有多种可能性，其中有一个会导致不被期望的灾难性结果，那么这个可能性就会发生”；
- 对于我们的代码来说——如果代码可能出现错误，它真的就会出现.....

```
int low_quality_programming(char *input) {  
    char str[1000 + 1]; // one more for the null character  
    strcpy(str, input); // copy input  
    ...  
}
```

```
int high_quality_programming(char *input) {  
    char str[1000 + 1]; // one more for the null character  
    strcpy(str, input, 1000); // copy input  
    str[1000] = '\\0'; // add terminating null character  
    ...  
}
```

- 01 | 别让代码附带有任何假设
- 02 | 没有被写下来的假设会导致各种问题
- 03 | 特别是代码规模变大的时候更是如此

- 01 | 检查所有外来数据的值
- 02 | 检查所有输入的参数
- 03 | 决定如何对不符合预期的输入进行处理

常见的检查项

- 检查确认所有的数组没有越界
- 解指针引用前确认他们都是非空的
- 确保函数的参数都是有效的
- 检查函数的返回值是符合预期的要求的
-

常见的处理方法

- 使用一个无害的值
- 使用最近一次的合法值
- 记录一个警告的日志
- 过程中抛出一个异常
- 进入一个专门的异常处理子程序
- 终止程序运行
-

好的工程师

01

关心代码的健壮性

02

对于代码中的每一个可能性都考虑得细致入微

03

对于奇怪的输入也都会给出解决的方案

04

确保自己的程序不会因为其他人的“愚蠢”而出现错误

差的工程师

01

假设自己的代码不会遇到会出现错误的情况

02

提交可能出错的代码，等着其他人以后来修

03

把各种假设放在自己脑子里，等待着被遗忘掉

04

产出不靠谱的软件代码

关于代码风格的几个问题

- 代码风格重要吗？为什么要考虑风格？
- 当我们说要考虑代码风格的时候，我们考虑的风格具体包括哪些部分？
- 代码风格的约束是否只能靠人？是否有更好的方式？



代码风格重要吗？为什么要考虑风格？

代码不仅仅是当下能用就好了.....

软件会发展就会需要人维护软件代码，那么就会有人阅读代码

代码是要给其他工程师读的

有好的风格的代码会最小化其他人阅读、理解代码所需要的时间

糟糕的代码风格

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include <time.h>
5  #define e(t, v, w) n t(u x, u y) { n a = sqrt(x+y); r ((v - a) * (w - a)); }
6  #define w while(
7  #define r return
8  typedef char u; typedef short q; typedef double n;
9  u *z = "Z\253\345\252\256\351\325\232\265f\256\246nyZf\253U^\266\232]z\352i\236\226i
[m\266\351\252\255\246\235V\247e\235\226\276Z\371jUg\236\212pX\254\251PH\253\332\225\373\235\b\v\336H0.
\203Qf\201\21d\207\211\357\367S7|)=\331n\24!E<\233Z\246\21flv\232d(r\263\274\375\313\305\200\361\302\327\274\375\300\201c\250-|\246]
\304\253I\377W\227R^\251\350\222C\251\203\3110\346\242\20dh\r\357\306x\325\274\315\366c\2th\325\272\376\311X+:\325\354q\263|
\257\21c\327\316g\204\266s\211\357\370A(\211""D\321\21d:\352\327\21bk@\211""E-\310s\271\235\304\273\22+\24I\274jT\256\363\214\307
{\232a\3270\2a+qm\270\2aI\315\337\220\217\218\315G\217\371@\215\335\v[\335\205\343\250-|\246\373:\225\2f\25520e)\206U\272\355\210\327)
u\rY]\276>\243\333w{3\219\2v""acC8\3775\351\b\377\204lJnsy.:\330\2553rKx'\362'!\%302\271\376|\236skq\242C\251\263&
\215K\24""fA\227\211\242r7]\371\333\200z%\277\207CJ\255^\264\237\327\275Z\n;
L\21aN#\371N\354G\256skq\242K!\364\211E\214\217\377\343z\340w\233\272\354eTD\262t\365\335[_\233\225J\313\2746\334\23\b\244F&\240]
\222\332\210\351\376rJs<\2265I;\203Xd\272\377\345\214\217\263]2I\237\302\253\346\355\332\343\350G\327\21350f}
\301n\223\366\300\210\343j\247\316%\363\210\345\n\252\215c\255\314\3248\300\261\350RM7\203\261\371\265\r_\220\217\219\2a(
+u\21d=I\235\314\331@\310C\216\210\307\345\357\306\21f""e\345R\354y\257\367\350!.\21f'\345uW&\204\246\367>\226\335\205\353\218\336;
\302\334\266\213Fn\233\201vAE\325+qnVR\222e\237E<e\203!f\333\345
[\217#\205\273I\314\371n\340\304\205\225h\323\336\371\21c""7\227\201%\327\201p\326\263\351f\219\2a\v\336;uI\312?\214\265I1$\302
+\373\233v\336\25\306,y\3130D\351d*\350f\357H\262]\23\332l\365L\237Qs\320{\231\244\204d\352v\335\215\327\357\200\247\302W\210", x, y;
10 n g() { r (n)rand()/18972; } n t, v; q F, B, C; e(a, 0, 3.0) e(b, 3.0, 4.125)
11 e(c, 4.125, 9) u f(n (*h)()) { do { t=g() * 6.283184; v = sqrt(g() * 72); y =
12 sin(t) * v; x=cos(t)*v; } w (h(x*x,y*y)>=0)|| (fabs(x) > 5) || (fabs(y) > 5));
13 r ((y+5) * 11) + x + 7; } u k(q x, u l) { u o = 0; F = 0; w F < 189) { B =C =
14 0; w 1) { y = ((z[F/4]>>((F++%4)*2))&3) - 1; if (y&2)
15 break; C = (C*2)+y, ++B; } if (!(C - x | B - l)) r (o<26)?o+0x61:((2108973>>(
16 (o-26)*8))&47); ++o; } r 1; } u s[1079]; u *E(u v) { u *x = s; w v--)w *(x++)
17 ); r x; } main() { q A = 0, B = 0, C = 1078; u l = 0; u *p = s; srand(time( 0
18 )); w C) { A = (A*2) + (((z[B/8+48]+114)^2)>>(7-(B++%8)))&1),++l; x = k[A,l
19 ]; if (x-1) { *(p++)=x, A=l=0, C--; } } printf("%s%s %s%s%s\n", E(0), E(f(c))
20 , E(f(a)), E(1), E(f(b)))); }

```

来源：国际C语言混乱代码大赛作品

代码风格的入门基础

01 | 命名

02 | 注释

03 | 格式

04 | 合理的组织

05 | 避免魔法数字

```
void checkForContinue(bool weShouldContinue)
{
    if (weShouldContinue) return;
}
```

命名 —— 符合语义

01 | 是什么

02 | 做什么

03 | 易于识别

命名 —— 符合语义

01

变量名

名词或名词化的动词，单复数应该正确

例如：apple, maxWindowWidth, selectedUsers

02

函数/方法名

动词 + 名词/形容词，考虑返回值类型

例如：countApples, isOccupied, increaseWindowHeight

命名 —— 符合语义

```
<el-collapse-item
  v-for="(item, index) of list"
  :key="index"
  v-bind:title=item.title
  v-bind:name=index
>
  <div>{{item.one}}</div>
  <div>{{item.two}}</div>
```

啥 list? 啥 item?

one 是啥? two 又是啥? i 是啥?

```
const i = {
  title: "欢乐斗bug",
  one:
    "与现实生活一致：与现实生活的流程、逻辑保持一致，遵循
    户习惯的语言和概念；",
  two:
    "在界面中一致：所有的元素和结构需保持一致，比如：设计
    式、图标和文本、元素的位置等。"
}

return {
  activeNames: ["1"],
  list: Array(10).fill(i)
};
```

命名 —— 清晰/避免缩写

01

不要有歧义

缩写或误导性的用词要避免，例如 Res 这种缩写、songCan 这种拼音

02

与类型对应

例如布尔类型的值可以用 isApple 或者 hasEars 却命名

03

避免无用单词

例如 all_of_the_students、myStudents 就不如直接些 students

04

前缀做命名指代的用法

例如 window-width 和 dialog-width 区分了是谁的宽度

命名 —— 清晰/避免缩写

res_backward

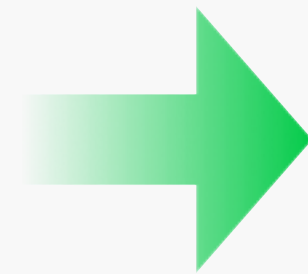
reaction

fanhuizhi

retval

return

myResponse



response

命名 —— 同类用途风格保持一致

01

大驼峰命名 (**upper camel case / pascal case**)

例如: Student, BestFriend

02

小驼峰命名 (**lower camel case**)

例如: student, bestFriend

03

下划线命名 (**snake case**)

例如: student, best_friend

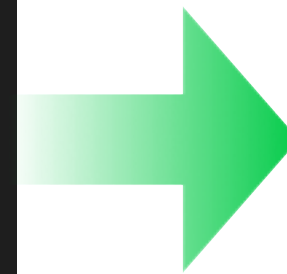
04

中划线命名 (**kebab case**)

例如: student, best-friend

命名 —— 同类用途风格保持一致

```
1  .Header {
2      text-align: right;
3      font-size: 18px;
4      padding-top: 15px;
5  }
6  #mainBody{
7      margin: 2px 20px;
8  }
9  #data_table{
10     margin: 30px 0px;
11 }
12 .menu-title{
13     font-size: 20px;
14     text-align: center;
15 }
```



```
1  .dialog-header {
2      text-align: right;
3      font-size: 18px;
4      padding-top: 15px;
5  }
6  .dialog-body {
7      margin: 2px 20px;
8  }
9  .dialog-inner-table {
10     margin: 30px 0;
11 }
12 .dialog-context-menu-title {
13     font-size: 20px;
14     text-align: center;
15 }
```


1

总结代码

为读你代码的人服务

2

解释为什么代码可行

解释一些奇技淫巧

3

标记一些内容

例如 Todo,

4

代码意图说明

指出解决的问题，而非方法

5

传达代码无法表述的信息

版权声明、版本号、保密要求等

6

重复注释

将代码实现的全过程都再描述一遍

1

总结代码

为读你代码的人服务

2

~~解释为什么代码可行~~

~~解释一些奇技淫巧~~

3

标记一些内容

例如 Todo,

4

代码意图说明

指出解决的问题，而非方法

5

传达代码无法表述的信息

版权声明、版本号、保密要求等

6

~~重复注释~~

~~将代码实现的全过程都再描述一遍~~

注释 —— 注释并不是越多越好

01

每一页显示的代码减少了

02

增加了需要维护的内容 —— 注释本身

注释 —— 自注解好于强行注释

```
class Person(object):  
    ...  
    get_age_year_month_days()  
    get_last_modified_year_month_days()  
    ...
```

```
class Model01(object):  
    ...  
    get_ageiymd() // get age in year, month, days  
    get_lmdiymd() // get last modified in year, month, days  
    ...
```

01 | 缩进

02 | 空格

03 | 换行

格式：缩进

```
double newton(double (*fp)(double), double (*fp_prime)(double)) {  
    double x = 1.5;  
    while (fabs(fp(x)) > EPSILON) {  
x = x - fp(x) / fp_prime(x);  
    }  
    return x;  
}
```

```
double newton(double (*fp)(double), double (*fp_prime)(double)) {  
    double x = 1.5;  
    while (fabs(fp(x)) > EPSILON) {  
        x = x - fp(x) / fp_prime(x);  
    }  
    return x;  
}
```

格式：空格

- 01 | 双目运算符前后
- 02 | 同基本的英文写作要求（行内逗号后，行内分号后）等
- 03 | 括号的前后是否换行要统一

格式：换行

- 01 | 不同逻辑的连续代码之间可以有一个空行
- 02 | 函数之间可以有一个空行
- 03 | 一些语言大括号之前是否换行要统一

- 01 | linting 工具（如 cpplint）
- 02 | 一种静态分析工具
- 03 | 对代码的格式等风格进行检查

风格检查工具

```
8 export default {
9   name: 'La
10  data () {
11    return {
12    }
13  }
14 }
```

Trailing spaces not allowed. eslint(no-trailing-spaces)
Peek Problem Quick Fix...

```
uploader.js
1:13 error 'angular' is not defined no-undef
1:28 error Strings must use doublequote quotes
7:15 error Strings must use doublequote quotes
7:28 error Strings must use doublequote quotes
7:34 error Missing "use strict" statement strict
9:20 error Expected '===' and instead saw '==' eqeqeq
10:17 error Expected '===' and instead saw '==' eqeqeq
14:20 error Strings must use doublequote quotes
27:16 error 'FormData' is not defined no-undef
34:17 error 'XMLHttpRequest' is not defined no-undef
35:1 error Trailing spaces not allowed no-trailing-spaces
36:12 error Strings must use doublequote quotes
45:1 error Unexpected blank line at end of file eol-last

â 13 problems
```

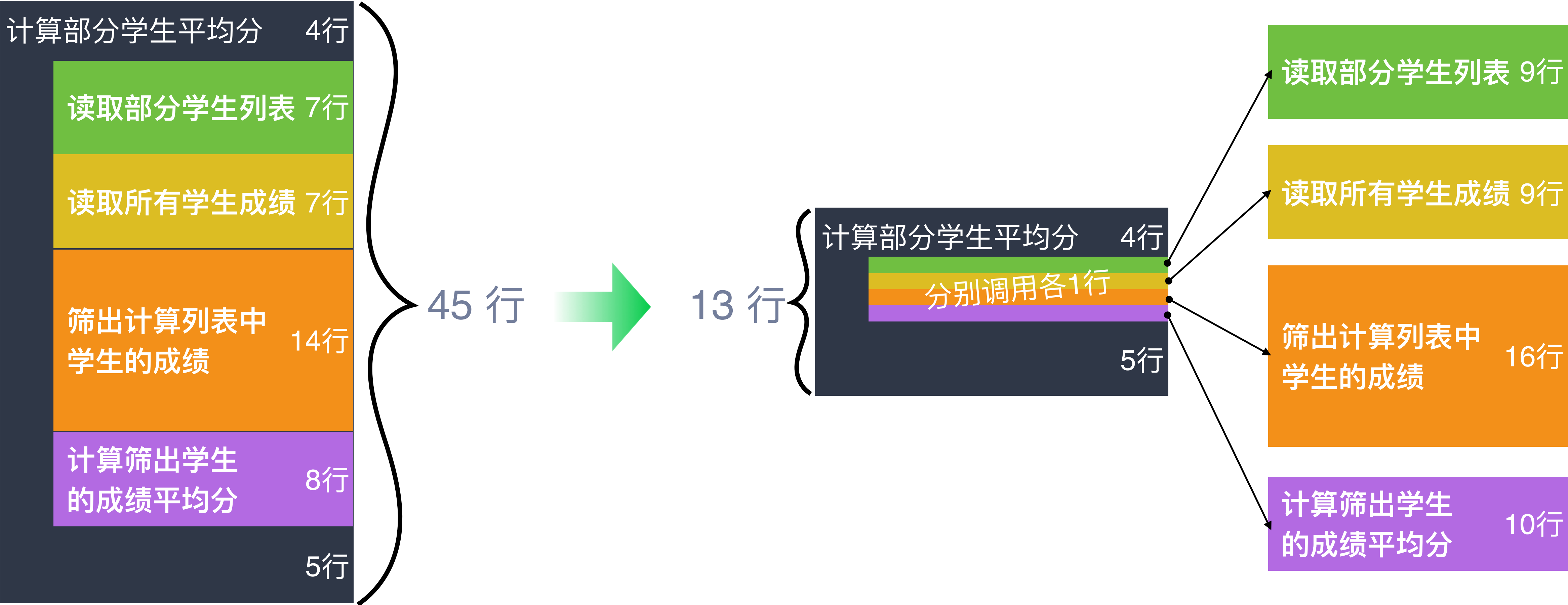

合理的组织

01 | 避免太长的子程序

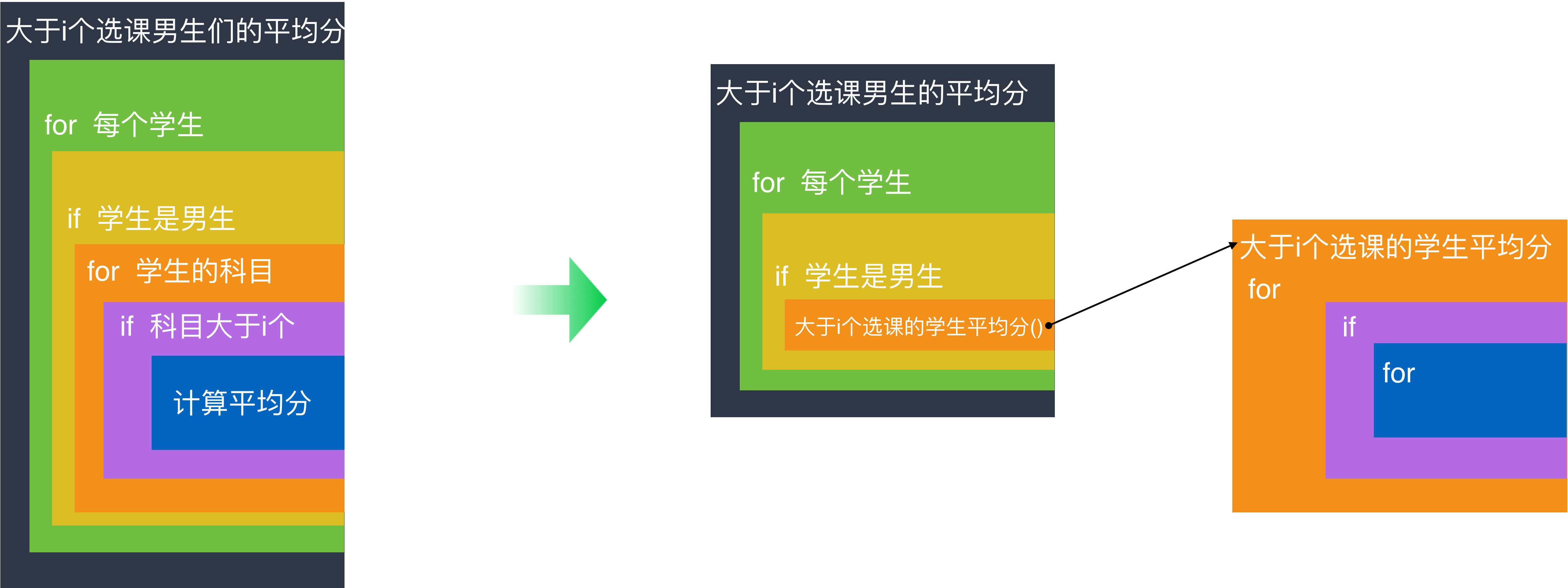
02 | 避免太深的嵌套

03 | 避免有太长的语句

合理的组织 —— 不要太长的子程序



合理的组织 —— 避免太深的嵌套



合理的组织 —— 避免有太长的语句

```
if (天气是晴天 或  
    ((天气是雨天 或 天气是雪天) 且 我带伞了) 或  
    (天气是沙尘天 且 我戴口罩了))
```

```
if (是可以出行的情况())
```

```
是可以出行的情况  
if (天气是晴天) {  
    返回 真  
} else if (天气是雨天 或 天气是雪天) {  
    if (我带伞了) {  
        返回 真  
    }  
} else if (天气是沙尘天) {  
    if (我戴口罩了) {  
        返回 真  
    }  
} else {  
    返回 假  
}
```

避免魔法数字

```
package Business;

import control.*;

public class Login3
{
    public int login32(String id,String pass)
    {
        Login4 cc = new Login4();
        if(cc.login43(id) == null)
            return 1;    //用户名不存在
        else
        {
            if(pass.compareTo(cc.login43(id).getpass()) == 0)//用户名匹配
            {
                if(cc.login43(id).getau() == 1)
                    return 3;    //管理员
                else
                    return 4;    //普通用户
            }
            else
                return 2;    //密码不正确
        }
    }
}
```

```
package control;

import frames.*;
import java.sql.*;

import userinterface.*;
import baseclass.LogInfo;

public class Login4
{
    public LogInfo login43(String id)
    {
        try{
            String sql = "select * from user where user

            ResultSet rs = DatabaseConnection.getDb().q

            if(rs.next())
```

```
package userinterface;

import Business.*;
import baseclass.*;

public class Login2 {
    public String id,pass;
    public int login21(LogInfo log){
        id = log.getuse();
        pass = log.getpass();
        if(id.compareTo("") == 0)
            return 1;    //还未输入用户名
        else if(pass.compareTo("") == 0)
            return 2;    //还未输入密码
        else
        {
            Login3 bb = new Login3();
            if(bb.login32(id, pass) == 1)
                return 3;    //用户名不存在
            else if(bb.login32(id, pass) == 2)
                return 4;    //密码不正确
            else if(bb.login32(id, pass) == 3)
                return 5;    //用户名匹配,管理员
            else
                return 6;    //用户名匹配,普通用户
        }
    }
}
```

- <https://google.github.io/styleguide/>
- <https://firefox-source-docs.mozilla.org/code-quality/coding-style/index.html>
- <https://github.com/search?q=style-guide&s=stars>
- <https://www.cs.umd.edu/~nelson/classes/resources/cstyleguide/>