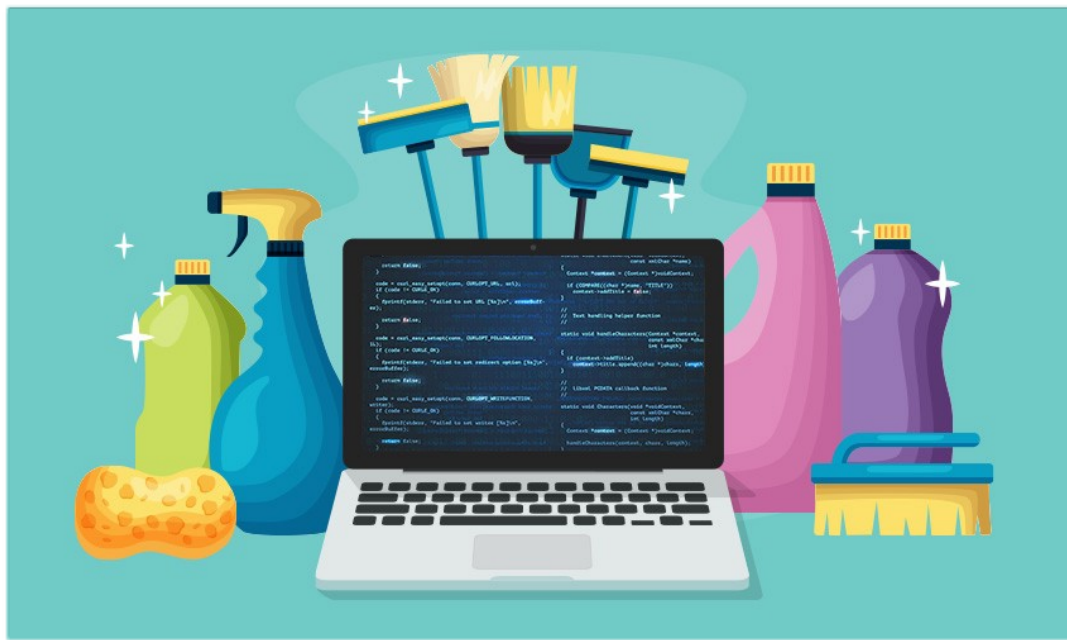


کد نامه

ویژه ی دانش جویان مبانی برنامه سازی نیم سال اول ۱۴۰۰-۱۳۹۹ دانشکده ی مهندسی کامپیوتر دانشگاه صنعتی شریف



در این شماره از
کدنامه، می خوانید:



کدهای مرتب و منظم



اشاره گرهای پرحاشیه!

کدهای مرتب و منظم

همیشه تمیز، مرتب و منظم کد بنویسید

هر برنامه، درست مانند یک ساعت است که از قطعات کوچک تر بسیاری تشکیل شده است، هم چون دستورات و عملیات، متغیرها، عبارات شرطی و حلقه ها، توابع و...؛ برای اینکه این ساعت بتواند به درستی و دقت کار کند، نیاز است که این قطعات با ظرافت و دقت بالا کنار یکدیگر قرار بگیرند. یکی از مهم ترین نکاتی که یک کد تمیز را از بقیه متمایز می کند، نحوه چیدمان و نظام دادن به این قسمت ها است، به طوری که ارتباط معنادار و صحیحی بین این اجزا برقرار شود. با کدنامه همراه باشید.

17 دی

ساختارها

آشنایی با راهی جدید برای کنترل و مدیریت داده های مرتبط در کنار یکدیگر

15 دی

ساختارها

آشنایی با راهی جدید برای کنترل و مدیریت داده های مرتبط در کنار یکدیگر

13 دی

ساختارها

آشنایی با راهی جدید برای کنترل و مدیریت داده های مرتبط در کنار یکدیگر

مطالب تدریسی شده

در کلاس درس

توسط استاد در

هفته ی گذشته

کدهای مرتب و منظم

متن داغیانی

- بعد از کاراکتر نقطه ویرگول در حلقه‌ها، یک فاصله می‌گذاریم، اما قبل از آن هیچ فاصله‌ای نباید وجود داشته باشد.

```
for (int i = 0; i < 10; i++) // valid
```

```
for (int i = 0;i < 10;i++) // invalid
```

چیدمان اجزای اصلی کد تمیز

جداسازی اجزا

استفاده از خطوط خالی برای جداسازی اجزای مختلف برنامه، راهکار هوشمندانه‌ای برای نشان دادن ارتباط آن‌ها با یک دیگر است. در تعریف توابع سعی کنید به کمک خطوط خالی ارتباط توابع را مشخص کنید. اگر گروهی از توابع هستند که در عملیات مشترکی مورد استفاده قرار می‌گیرند (مثلاً تمام توابعی که مربوط به پیاده‌سازی بخش شبکه‌ی پروژه هستند)، بهتر است آن‌ها را پشت سر هم تعریف کرده و بین‌شان ۲ خط خالی قرار دهید. حال برای مجزا کردن این توابع از گروهی دیگر کافیت بین آن‌ها از تعداد خطوط خالی بیشتری استفاده کنید. هم‌چنین توابعی که به یکدیگر وابسته بوده و یکدیگر را در مراحل پیاده‌سازی خود صدا می‌کنند را نیز بهتر است مشابه توضیحات بالا از دیگر قسمت‌ها تفکیک کنید.

هرچه نزدیک‌تر، بهتر

سعی کنید متغیرهای محلی را درست قبل از اولین استفاده آن‌ها تعریف کنید. با تعریف یک‌باره تمامی این نوع متغیرها در ابتدای یک بلاک، ممکن است در ادامه نوع آن‌ها یا علت تعریفشان را فراموش کنید و نیاز باشد تا مکرراً به ابتدای بلاک بازگردید. هم‌چنین اگر یک تابع، تابعی دیگر را صدا می‌زند، تابع صداکننده باید درست پایین تابع صداشونده قرار گیرد. رعایت این نکته کمک می‌کند تا کد یک رفتار و جریان طبیعی را دنبال کند.

اشاره‌گرهای پر حاشیه (قسمت دوم)

علی حاتمی

در شماره‌ی پیشین کدنامه، با اشاره‌گرها و کاربردهای دسترسی به آدرس متغیرها در حافظه، آشنا شدیم.

در این مقاله، به چند کاربرد و نکته‌ی دیگر در خصوص اشاره‌گرها می‌پردازیم.

چاپ درست آدرس اشاره‌گرها

در این مقاله، خواهیم دید که چگونه می‌توانیم کدمان را مرتب کنیم تا ساختار منظم‌تری پیدا کند.

فاصله‌ها را رعایت کنیم!

خطوط نه چندان بلند

طولانی بودن بیش از اندازه خطوط باعث می‌شود تا خوانایی کد به شدت کاهش یابد. طول هر خط بهتر است حداکثر به اندازه ۸۰ تا ۱۲۰ کاراکتر باشد. این مقدار متن در اکثر نمایشگرها و برنامه‌های نگارشی در یک صفحه قابل نمایش است.

کاراکتر Tab

در اکثر برنامه‌های ویرایش متن، امکان تغییر طول کاراکتر Tab بر حسب تعداد فاصله (space) وجود دارد. این مقدار معمولاً ۴ یا ۸ فاصله در نظر گرفته می‌شود. توجه کنید که در تمام کدتان از یک فاصله مشخص استفاده شده باشد تا تراز آن در تمامی خطوط یکسان و مشخص باشد.

کاراکترهای متشخص

هر برنامه‌ای پر از دستورات و عملیات ریاضی و کاراکترهای غیر حرفی است که موارد زیر به افزایش خوانایی آن‌ها کمک می‌کند:

- همواره قبل از باز کردن پرانتز و بعد از بستن آن یک فاصله قرار دهید؛ اما میان پرانتزها با عبارت داخلی آن نباید هیچ فاصله‌ای وجود داشته باشد.

```
if (a == 2) // valid
```

```
if(a == 2) // invalid
```

```
if (a==2) // invalid
```

- بهتر است همواره قبل و بعد از عملگرهای ریاضی یا دودویی از یک فاصله استفاده کنید. (البته به جز عملگرهایی مانند ++)

```
a = b + c; // valid
```

```
a=b+c; // invalid
```

• `[x]` یا `[]` جای‌گزین شود با «آرایه به اندازه `x` از ... آرایه با اندازه نامشخص از...»

• `*` جای‌گزین شود با «اشاره‌گر(هایی) به ...»

۲. این کار را در جهت ماریپیچ و ساعت‌گرد، تا زمانی که تمام اجزا پوشش داده شوند، انجام دهید.

مثال: عبارت زیر به چه معناست؟

```
char * var [10];
```

پاسخ:

```

+-----+
|  +-+  |
|  ^   |
char *var[10];
|  ^   |
|  +---+
+-----+

```

اول از کلمه `var` شروع می‌کنیم: «`var` ... است.»

سپس ساعت‌گرد به راست می‌رویم تا به `[10]` برسیم: «`var` یک آرایه به اندازه‌ی ۱۰ از ... است.»

سپس می‌چرخیم تا به `*` برسیم: «`var` یک آرایه به اندازه‌ی ۱۰ از اشاره‌گرهایی به ... است.»

سپس می‌چرخیم و یک دور کامل دور می‌زنیم تا به `char` برسیم:

«`var` یک آرایه به اندازه‌ی ۱۰ از اشاره‌گرهایی به نوع داده‌ی `char` است.»

در نتیجه، `var` یک آرایه است که ۱۰ عضو دارد و هر عضو آن، خود شامل آدرس یک خانه‌ی حافظه است که در آن خانه، یک کاراکتر ذخیره شده است. می‌توانید با روش فوق، به همراه برخی مثال‌های پیچیده‌تر، در این لینک آشنا شوید.

همان‌طور که در کدنامه‌ی شماره‌ی قبل دیدید، با چاپ به کمک `%d` و به دلیل سیستم مکمل دو، ممکن است عدد نامربوطی هنگام چاپ آدرس‌ها به دست آید؛ در نتیجه بهتر است از `%p` به جای `%d` استفاده کنید.

آدرس توابع

در کامپیوتر می‌توان اشاره‌گرهایی تعریف کرد که به جای اشاره به یک متغیر، به یک تابع اشاره کنند؛ به مثال زیر دقت کنید:

```

void myfunc (int a) {

    printf("Parameter 'a' is %d\n", a);

}

int main () {

    // fun_ptr is a pointer to function

    void (*func_ptr) (int) = &myfunc;

    // Calling foo () using fun_ptr

    (*func_ptr) (10);

    return 0;

}

```

خروجی کد فوق، `Parameter 'a' is 10` خواهد بود. همان‌طور که در مثال بالا می‌بینید، `func_ptr` یک اشاره‌گر به تابع `myfunc` است. شاید از خود بپرسید که چنین فرآیندی چه کمکی به ما خواهد کرد؟

از کاربردهای مهم این نوع اشاره‌گرها، پاس دادن یک تابع به تابع دیگر است که نمونه‌ای از آن را هنگام بررسی تابع مرتب‌سازی سریع در زبان C در کلاس درس خواهید دید.

تحلیل عبارات پیچیده‌ی ترکیبی

گاهی هنگام خواندن برنامه‌ها، به تعاریف پیچیده‌ی مرکب از نمادهای گوناگون بر می‌خورید؛ می‌توانید از روش زیر برای یافتن معنی عبارات پیچیده استفاده کنید:

۱. با شروع با اسم متغیر، در جهت ماریپیچ و ساعت‌گرد حرکت کنید. هنگام مواجهه با عناصر زیر، عبارات مربوطه را جایگزین آن‌ها کنید:

ارتباط با کدنامه



خوش‌حال می‌شویم اگر پیشنهادات و انتقاداتی نسبت به کدنامه دارید یا سوال خاصی دارید که تمایل دارید در کدنامه پاسخ داده شود، به دستیاران آموزشی درس اطلاع دهید.