

7 راه پیشنهادی برای پرهیز از کدنویسی احمقانه و آماتور

این مقاله را به صورت تحلیلی و در سطح متوسط به بالا می نویسم و مخاطب بنده کاربران جدید نیوک لرن نیست پس اگر چیزی نفهمید مانند نویسنده آن هستید که خودش هم نفهمید می خواهد چه بنویسد .

در توسعه نرم افزاری و پروژه ای اگر خاصیتی هم بگویم نداشته باشد (ارتقا دانش توسعه دهنده) یک تجربه خوبی است برای آشنا شدن با کدنویسی و کدهای احمقانه و ضعیف . امیدوارم در برابر واژه احمقانه جبهه گیری نشود . چون واقعا برخی کدها ضعیف و بی فکر نوشته می شوند .

در این مطلب 7 روش پرهیز از این دسته کدها را ذکر خواهیم کرد

1- استفاده از ساختار شی گرا OOP که به قوائد کدنویسی OO نیز مشهور است .

یک کدنویس آماتور اولین کاری که می کند تامین هدف کوتاه مدتش است . یعنی انجام شدن آن هدف و کدهای ابتدای بدون توجه به توسعه آینده .

اسکرپت (نرم افزار) را اگر یک ساختمان بگیریم . معمار بی دانش با توجه به پولی که در دست دارد شروع می کند به ساختن یک خونه (تپه گل) کاه گلی (آجری) که نه بر اساس یک پروسه علمی بنا شده است و نه مقاومت خوبی دارد و در صورتی که بخواهد در آینده آن را ارتقا دهد همه اش باهم می ریزد و می بایست از ابتدا بسازد . قصیه استفاده از اصول OO هم در برنامه نویسی همین است . کلاس نویسی و شی گرایی به قول برنامه نویسان شالوده و مایه یک اسکرپت حرفه ای است . ابزاری که در آینده هر لحظه بخواهد می تواند با پیوند به آن بخش کدهایش را توسعه دهد .

نمونه بارز عدم استفاده از ساختار شی گرا و کلاس نویسی دقیقا همین پرتال نیوک است که بر اساس مایحتاج کنونی و هدف نخستین یک ماژول شروع به کار می شود و خدا می داند چه خواهد شد .

پس اولین کار نوشتن یک کلاس مرتبط با کار ما است و سپس فراخوان آن کلاس و به راحتی نوشتن کدهای منظم .

2- بازنگری , بازنگری و در نهایت هم بازنگری

یک اصل مهم در برنامه نویسی همین قسمت است . بازنگری را معادلی برای واژه code refactoring بکار بردم . هدف این بخش بازنگری و ارتقا قسمت های کنونی اسکرپت است . حتما در طول توسعه یک بخش دقت می کند و آن بخش را از لحاظ فنی ارتقا می دهد . این یک اصل است و برنامه نویسان حرفه ای کدهای خود را مجدد بررسی می کنند و تا آنجا که می توانند ساختار شکنی می کنند و در این بین ممکن است فکر کنید زمان زیادی را از دست می دهید و باید به شما بگویم سخت در اشتباهید چون زمان زیادی را بدست می آورد از آنجایی که از تضعیف نرم افزار جلوگیری می کنید .

3- هرچه ساده تر , بهتر

برخی ها فکر می کنند کدها باید در هم تنیده باشن , آنقدر که اگر کسی به کدهای ما نگاه کرد سردرد بگیرد و

عطایش را به لقایب ببخشید . کدهای اسکریپت های حرفه ای ساده و مرجع گرا هستند . نیازی نیست کدی را که در یک بخش از نرم افزار نوشتید تکرار کنید و با یک ساختار ساده فراخونی می توانید از این جرکت ابتدایی جلوگیری کنید . پس از پیچیده کردن بی جهت کدها خودداری کنید .

4- از الگوهای کدنویسی استفاده کنید

زمانی که الگوهای جهانی برای کدهای خود استفاده می کنید این شانس را دارید که در زمان برخورد با مشکل بتوانید آن را گوگل کنید و امیدوار باشید تا به جواب سوال خود برسید و از وقت گذاشتن بیهوده روی حل مشکل خودداری کنید .

الگوهای طراحی (کدنویسی) را می توان از کتاب ها یا سایت های مرجع آن زبان برداشت کرد که از کتابخانه های لازم برای کدنویسی برخوردارند .

5- پذیرفتن محدودیت زبان برنامه نویسی

خیلی از زبان های برنامه نویسی دارای محدودیت هایی هستند و واقعا برخی اوقات برفرض توقع بیش از حدی از زبان مثلا PHP می رود . به اصلاح برخی ها که حرفه ای تر هستند سعی در اصلاح آن محدودیت ها دارند و این کار کلا شخصیت و هویت آن زبان را زیر سوال می برد و برخی اوقات اوضاع بدتر از قبل می شود و کلا افتضاحی به بار می آید که راه برگشتی نیست .

6- از بارخوردها و نظرات شخص دومی استفاده کنید

یکی از خوبی های کار گروهی وجود چند تفکر است که گاه منجر به پیشرفت آن اسکریپت می شود (البته نه در ایران) Peer کلمه ای است که به همکار - دستیار - مشاور و حتی شخص دوم گفته می شود و در کدنویسی می تواند شما را بسیار کمک کند

7- توپ دوست شماست

این عنوان برای یک مقاله کدنویسی احمقانه است اما شما به جای کلمه توپ واژه کدنویسی PHP اش را بگذارید

. است شما دوست E_STRICT

این به معنای آن است که سعی کنید همواره این امکان روشن باشد و خطاهای موجود در صفحه را مشاهده کنید . دقت داشته باشید که این امکان بسیار به نسخه php شما بستگی دارد که معمولا خطاهای نمایش داده شده بستگی به نسخه ابزار شما دارد .

مطمئنا موارد بهتر و بیشتری هم هست که فرصت آن در این مقاله خلاصه وجود ندارد

موفق باشید

