به نام خدا

نام و نام خانوادگی: امیرحسین نظری یامچلو

شماره دانشجویی: 02121033720039

نام درس: برنامه نویسی سمت سرور

نام استاد: آقای میثاق یاریان

موضوع: تحقیق راجب اصل های yagni,kiss,dry,solid و deconstructor, constructor,gc.collect

تمرین شماره 5

KISS

“Keep It Simple, Stupid”

در برنامه نویسی بسیار اهمیت دارد.سعی کنید این اصل را سعی کنید به خاطر بسپارید و KISS اصل برای حفظ آن تلاش کنید.هرچقدر کد ساده تر باشد نگهداری آن در آینده ساده تر است.افراد دیگری که .بخواهند کد شما را مورد ارزیابی قرار دهند در آینده با استقبال بیشتری این کار را انجام میدهند

پایه گذاری شده است و سیستم های خوب به جای پیچیدگی، به Kelly Johnson توسط KISS اصل سمت ساده سازی پیش میروند.از این رو سادگی، کلید طلایی طراحی است و باید از پیچیدگی های غیرضروری دوری کرد.

YAGNI

”You Aren’t Gonna Need It“

گاهی اوقات تیم های توسعه و برنامه نویسان در مسیرپروژه تمرکز خود را بر روی قابلیت های اضافه ی پروژه که "فقط الان به آن نیاز دارند" یا "در نهایت به آن نیاز پیدا میکنند" میگذارند.در یک کلام :اشتباه است!در اکثر مواقع شما به آن نیاز پیدا ندارید و نخواهید داشت. "شما به آن نیازی ندارید".

اصل YAGNI قبل از کدنویسی بی انتها و بر پایه ی مفهموم "آیا ساده ترین چیزی است که می تواند احتمالا کار کند" قرار دارد.حتی اگر YAGNI را جزوی از کدنویسی بی انتها بدانیم، بر روی تمام روش ها و فرآیند های توسعه قابل اجرا است.با پیاده سازی ایده‌ی "شما به آن نیازی ندارید" میتوان از هدر رفتن وقت جلوگیری کرد و تنها رو به جلو و در مسیر پروژه پیش رفت.

DRY

”Don’t Repeat Yourself“

تا الان چندین بار به کد های تکراری در پروژه برخورد کرده اید؟ اصل DRY توسط Andrew Hunt و David Thomas در کتاب The Pragmatic Programmer پایه گذاری ده است.خلاصه ی این کتاب به این موضوع اشاره میکنید که "هر بخش از دانش شما در پروژه باید یک مرجع معتبر، یکپارچه و منحصربفرد داشته باشد". به عبارت دیگر شما باید سعی کنید رفتار سیستم را در یک بخش از کد مدیریت کنید.

از سوی دیگر زمانی که از اصل DRY پیروی نمیکنید، در حقیقت اصل WET که به معنای Write Everything Twiceیا We Enjoy Typing دامن گیر شما شده است! (لذت بردن از وقت تلف کردن)

اصول SOLID در برنامه نویسی چیست؟

اصطلاح SOLID اولین بار توسط مایکل فیرز معرفی شد، در حالی که خود اصول در ابتدا توسط رابرت جی. مارتین، همچنین به نام عمو باب، در مقاله خود در سال 2000 ارائه شد. عمو باب دانشمند کامپیوتر مشهور، نویسنده کتاب‌های پرفروشی مانند «Clean Code» و «Clean Architecture» و یکی از شرکت‌کنندگان فعال در Agile Alliance است.
اصول SOLID در برنامه نویسی با مفاهیم کدنویسی تمیز، معماری شی گرا و الگوهای طراحی همسو هستند، زیرا همگی هدف مشترک ایجاد نرم‌افزار با کیفیت بالا را دارند.

در اصل SOLID از ۵ اصل اساسی تشکیل شده است که به صورت موارد زیر هستند:

اصل مسئولیت واحد (Single Responsibility Principle)

اصل باز – بسته (Open-Closed Principle)

اصل جایگزینی لیسکوف (Liskov Substitution Principle)

اصل جداسازی رابط (Interface Segregation Principle)

اصل وارونگی وابستگی (Dependency Inversion Principle)

اصل مسئولیت واحد در اصول SOLID در برنامه نویسی

اصل مسئولیت واحد (SRP) تأکید می‌کند که یک کلاس باید تنها یک دلیل برای تغییر داشته باشد. اصل مذکور این ایده را ترویج می‌کند که هر کلاس باید یک مسئولیت واحد داشته باشد و تنها یک مشکل را حل کند. با رعایت این اصل، تست، نگهداری و درک کد آسان‌تر می‌شود.

استفاده از SRP در کلاس‌ها، اجزای نرم‌افزار و میکروسرویس‌ها به طراحی بهتر نرم‌افزار منجر شده و به جلوگیری از عوارض جانبی ناخواسته هنگام ایجاد تغییرات کمک می‌کند.

---

deconstructor

بله، یک deconstructor یا destructure یک عملگر در زبانهای برنامه نویسی مانند JavaScript و ECMAScript 6 است که اجازه می‌دهد تا مقادیر از داخل یک آرایه یا شیء را دریافت کرده و آن‌ها را به متغیرهای جداگانه انتقال دهیم. به عبارت دیگر، deconstructor به برنامه نویس اجازه می‌دهد تا یک آرایه یا شیء را به اجزای آن تجزیه کند. این ویژگی مفید است زیرا برنامه نویس می‌تواند به راحتی به اجزای یک آرایه یا شیء دسترسی پیدا کند و از آن‌ها استفاده کند.

مثال:

```
// با استفاده از deconstructor یک آرایه را تجزیه می‌کنیم
let arr = [1, 2, 3];
let [a, b, c] = arr;
console.log(a); // 1
console.log(b); // 2
console.log(c); // 3

// با استفاده از deconstructor یک شیء را تجزیه می‌کنیم
let obj = { x: 1, y: 2, z: 3 };
```

```
;let { x, y, z } = obj

console.log(x); // 1

console.log(y); // 2

console.log(z); // 3
```

constructor

کاملاً درست! constructor در برنامه‌نویسی به معنای یک متد ویژه است که هنگام ایجاد یک شیء یا نمونه جدید از یک کلاس صدا زده می‌شود.

constructor معمولاً برای مقداردهی اولیه اشیاء و انجام تنظیمات اولیه مورد نیاز برای ایجاد نمونه‌های جدید از یک کلاس استفاده می‌شود.

در زبان‌های برنامه‌نویسی مختلف از جمله JavaScript، Java، C++ و... constructor به صورت خاصی ممکن است تعریف شود.

به عنوان مثال، در JavaScript، constructor با استفاده از کلمه کلیدی constructor تعریف می‌شود و در Java constructor با نام کلاس تعریف می‌شود.

در صورتی که به بیشتر یک زبان خاص نیاز دارید، ممکن است بتوانم اطلاعات بیشتری راجب تعریف constructor در آن زبان ارائه دهم. 🛠

gc.collect

gc.collect یک تابع در زبان برنامه‌نویسی پایتون است که برای اجباری اجرای مجدد ماشین گرفته‌بندی (garbage collection) در فضای حافظه استفاده می‌شود.

وظیفهٔ gc.collect این است که اشیاءی که دیگر مورد استفاده نیستند را از حافظه حذف کند.

وقتی از این تابع استفاده می‌کنید، بدنهٔ جایگزین مجموعه‌ای از شی‌های غیرقابل دسترس موجود در حافظه اجرا می‌شود.

این کار ممکن است منجر به آزادسازی حافظه غیرضروری و افزایش عملکرد برنامه شما شود.

استفاده‌ی درست از gc.collect برای بهینه‌سازی مصرف حافظه و جلوگیری از نشتی حافظه بسیار حیاتی است، اما باید با احتیاط و هماهنگی با نیازهای واقعی برنامه شما انجام شود. 🧹