

به نام خدا

تحقیق

نیم سال ۱۴۰۲۱

توضیحات

- به علت دیر منتشر شدن تحقیق، اگر کلاً تحقیق را انجام ندهید، دو نمره‌ی تحقیق را دریافت می‌کنید! چنانچه می‌خواهید تحقیق را انجام دهید، ادامه‌ی این مستند را بخوانید...
- لطفاً نتیجه‌ی تحقیق را حداکثر تا ساعت ۲۳:۵۹ تاریخ تعیین شده در صفحه‌ی درس‌افزار درس بارگذاری نمایید.
- توجه کنید که هر تیم باید یک تحقیق را انتخاب کند. یک تحقیق می‌تواند توسط چند تیم انتخاب شود.
- ذکر **نام و نام خانوادگی** به همراه **شماره دانشجویی** همه‌ی اعضای گروه، هم‌چنین **شماره‌ی تیم** در فایل PDF نتیجه‌ی تحقیق ضروری است. در صورتی که نام هر یک از اعضای گروه در فایل نتیجه‌ی تحقیق نباشد، به منزله عدم همکاری آن عضو در گروه و نارضایتی سایر هم‌گروهی‌ها محسوب شده و نمره تحقیق برای آن فرد لحاظ نخواهد شد.
- چنانچه یک نفر از اعضای هر گروه نتیجه‌ی تحقیق را در درس‌افزار درس بارگذاری کند، کافی است.
- نتیجه‌ی تحقیق را به زبان **فارسی** بنویسید. در صورتی که ترجمه‌ی کلمه‌ای ناملموس می‌شد، واژه‌ی اصلی را به صورت پانویس اضافه کنید.
- تحقیق از دو نمره است و یک نمره‌ی امتیازی دارد؛ اگر نتیجه‌ی تحقیق خود را به صورت ساختارمند و با رعایت اصول نگارشی مناسب بنویسید، ۰.۲ نمره‌ی امتیازی دریافت می‌کنید. هم‌چنین اگر نتیجه‌ی تحقیق خود را به صورت یک مقاله‌ی باکیفیت در بستر وی‌رگول منتشر کنید، ۰.۴ نمره‌ی امتیازی دیگر دریافت می‌کنید. به علاوه، اگر با نتایجی که از تحقیق خود به دست آورده‌اید، به تکمیل حداقل دو مقاله‌ی مرتبط در ویکی‌پدیا بپردازید، ۰.۴ نمره‌ی امتیازی دیگر دریافت خواهید کرد.

موفق باشید

تیم آموزش مهندسی نرم‌افزار

sharif.software.engineering@gmail.com

تحقیق یک: چارچوب‌های اصول‌مند

مفاهیم حوزه‌ی دانش مهندسی نرم‌افزار را می‌توان به چهار دسته‌ی زیر تقسیم کرد:

1. اصل (principle): اصل‌ها مفاهیم درشت‌دانه‌ای هستند که با گذر زمان بسیار کم تغییر می‌کنند. هم‌چنین تمام حوزه‌ی دانش مهندسی نرم‌افزار بر آن‌ها بنا شده است؛ برای مثال انتزاع (abstraction)، استفاده‌ی مجدد (reuse)، چسبندگی (cohesion)، جفتگری (coupling) و اصول SOLID، چند اصل مهم در مهندسی نرم‌افزار هستند.
2. رویه (practice): رویه‌ها، خط‌مشی‌هایی هستند که کارابودن آن‌ها با توجه به تجربیات قبلی در شرایط مشخصی نشان داده شده است؛ برای مثال «توجه به چిستی تا چگونگی در جمع‌آوری نیازمندی‌ها» یک رویه است.
3. فن (technique): فن‌ها روش‌های حل مسئله‌ای هستند که در فرآیند ایجاد نرم‌افزار استفاده می‌شوند؛ مانند آزمون واحد (unit testing) و مدل‌سازی مورد استفاده (use case modeling).
4. ابزار (tool): ابزارها محصولات به‌روزی هستند که استفاده از اصول و رویه‌ها را آسان می‌کنند.

چارچوب‌های برنامه‌سازی (مانند React یا Django) در دسته‌ی ابزارها قرار می‌گیرند. همانطور که در تعریف ابزار است، این چارچوب‌ها استفاده از اصول یا رویه‌ها را آسان‌تر می‌کنند.

به انتخاب خود، دو چارچوب برنامه‌سازی سطح بالا را انتخاب کنید و برای هر چارچوب، ده مورد از اصول یا رویه‌هایی را که این چارچوب باعث رعایت آن‌ها می‌شود، با ذکر مثال (قطعه کد، مدل‌سازی UML، سناریوی نیازمندی و ...) توضیح دهید.

به‌عنوان نمونه، ساختار مولفه‌ای (component structure) در React باعث افزایش قابلیت reliability می‌شود. در موردی دیگر، hook ها در React به‌نوعی جایگزین ارث‌بری (inheritance) در مولفه‌های تابعی (function components) هستند.

تحقیق دوم: مصاحبه

آیا واقعاً درس مهندسی نرم‌افزاری که گذرانیدیم، به درد می‌خورد؟ برای بررسی این موضوع، می‌توان سراغ دو نفر از اعضای فنی یک شرکت رفت و با مصاحبه با آن‌ها، در مورد کارکرد درس مهندسی نرم‌افزار اطلاعاتی را به دست آورد. توجه کنید که این اعضای فنی نباید تازه‌کار (junior) یا از دانشجویان فعلی درس باشند.

مواردی که در این تحقیق باید تحویل دهید، عبارتند از:

- اطلاعات افراد مورد مصاحبه (شامل نام و نقش آن‌ها در شرکت)
- فایل صوتی مصاحبه‌ها
- گزارش مکتوب مصاحبه‌ها
- پنج مورد از کاستی‌های درس که از صحبت با افراد به دست آورده‌اید
- پنج مورد از نقاط قوت درس که به آن افراد در کارشان کمک می‌کند

تحقیق سوم: ویکی‌پدیا

ویکی‌پدیا یکی از منابع آموزشی فارسی است. متأسفانه مقاله‌های مرتبط با حوزه‌ی دانشی مهندسی نرم‌افزار در ویکی‌پدیا وضعیت خوبی ندارند. در این تحقیق می‌خواهیم تا مقاله‌های زیر در حوزه‌ی مهندسی نرم‌افزار را ویرایش علمی و ادبی کنیم:

- [مهندسی نرم‌افزار](#)

- [زبان مدل‌سازی UML](#)

- [نمودار کلاس](#)

- [نمودار توالی](#)

هم‌چنین می‌خواهیم تا مقاله‌های زیر را ایجاد کنیم:

- [جریحه‌ی ایجاد نرم‌افزار](#)

- [مهندسی نرم‌افزار شی‌گرا \(OOSE\)](#)

برای ویرایش یا ایجاد بهتر مقاله‌ها، [اینجا](#) و [اینجا](#) را مطالعه کنید.

توجه کنید که اولویت اول، چه در ویرایش و چه در ایجاد مقاله، کیفیت علمی است. مقاله‌ها باید با علمی که در درس فراگرفته‌ایم، منطبق باشند. هم‌چنین، ساختار درست محتوای یک مقاله (بخش‌ها و زیربخش‌ها)، استفاده از معادل‌های مناسب فارسی و ارجاع به پیوندهای مناسب، اهمیت دارد.

این تحقیق از سه نمره است و نمره‌ی امتیازی ندارد. چنانچه می‌خواهید این تحقیق را انتخاب کنید، به ایمیل درس ایمیل بزنید. تحقیق به اولین تیمی که ایمیل بزند، می‌رسد.

تحقیق چهارم: Django در DDD

پیش‌نیاز:

● Django

در این تحقیق قصد دارید یک راهنمای سبک¹ برای اعمال Domain-Driven Design در Django - فناوری معروف برای ایجاد بک‌اند - را بشناسید. ابتدا این [راهنمای سبک](#) را مطالعه کنید سپس آموخته‌های خود را در قالب یک تحقیق علمی ارائه دهید. در متن تحقیق خود اطمینان حاصل کنید که موارد زیر در نظر گرفته شود:

1. تحقیق را با مقدمه‌ای بر معماری فعلی جنگو آغاز کنید.
2. توضیح دهید معماری فعلی جنگو چه ایراداتی داشته که این راهنمای سبک در پی حل آن‌ها است.
3. در بخشی از تحقیق توضیحاتی دهید که پیش‌زمینه‌ی درک Domain-Driven Design باشد. حتماً مفاهیم زیر را از صفحه‌ی اینترنتی فاولر مطالعه کنید و توضیحاتی از آن‌ها و ارتباطشان با این راهنمای سبک را ذکر کنید:

a. [Domain Model](#)

b. [Ubiquitous Language](#)

c. [Multiple Canonical Models](#)

d. [Bounded Context](#)

4. توضیحات کافی‌ای از دریافت خود و نحوه اعمال این راهنمای سبک ارائه دهید. این درک شما بالطبع نباید ترجمه‌ی کلمه به کلمه‌ای از متن خود راهنمای سبک باشد.
5. مدل‌های بصری استفاده شده در صفحه‌ی این راهنمای سبک به صورت غیررسمی هستند و از زبان خاصی پیروی نمی‌کنند. از محتوای درس آموختید که زبان مدلسازی که در عمل استاندارد شده است UML است. در مورد «نمودار مولفه»² - که برای مدلسازی معماری استفاده می‌شود - مطالعه کنید، علی‌الخصوص کاربرد Interface و نحوه استفاده درست از آن در این نمودار. سپس مدل‌های بصری از نوع نمودار مولفه در تحقیق خود قرار دهید که توضیحاتتان را کامل کند.

¹ Styleguide

² Component Diagram

تحقیق پنجم: یک Microservice کوچک

پیش‌نیاز:

● Python

در این تحقیق قصد دارید یک کتابخانه برای پیاده‌سازی یک پروژه Microservice کوچک را مطالعه کنید. از طریق وبسایت [Nameko](#) می‌توانید معرفی آن را مطالعه کرده و به صفحات Documentation و Github آن دست پیدا کنید. در نهایت آموخته‌های خود را در قالب یک تحقیق علمی ارائه دهید. در متن تحقیق خود اطمینان حاصل کنید که موارد زیر در نظر گرفته شود:

1. مقدمه‌ای از عملکرد سیستم‌های با معماری Microservice که از محتوای درسی آموختید ارائه دهید.
2. توضیح دهید (۱) به چه شکل و (۲) تا چه حد این کتابخانه پایبند به اصول این معماری است.
3. این کتابخانه بر پایه الگوی Dependency Injection راه‌اندازه شده است. بر اساس [وبگاه فاولر](#) توضیح دهید که این (۱) الگو چگونه کار می‌کند و (۲) چگونه در این کتابخانه از آن بهره‌برداری شده است.
4. در موارد ۲ و ۳، حتماً از مقداری مدلسازی UML («نمودار کلاسی» و «نمودار مولفه») و قطعه کدهایی که توضیحاتتان را کامل کند بهره ببرید.

در نهایت به سوال زیر در تحقیق خود پاسخ دهید:

ابتدا اصل [DIP](#) از اصول SOLID را به دقت مطالعه کنید. حال فرض کنید یک سرویس ریزدانه، به نام A، در میکروسرویس خود داریم که تنها راه ارتباط با آن از طریق یک کلاس [Facade](#) است که از کلاس‌های Service درون کتابخانه Nameko ارث‌بری می‌کند و AServiceFacade نام دارد. این کلاس تعدادی متود از جنس [RPC](#) دارد. سرویس B از میکروسرویس‌های موجود، کارخواه این سرویس است. در حال حاضر بدون استفاده از DIP، سرویس B، سرویس A را درون خود نصب می‌کند تا از متودهای AServiceFacade مطلع شود و از آن‌ها استفاده کند - در این سامانه، همه سرویس‌ها با استفاده از `setuptools` تبدیل به پکیج‌های قابل نصب شده‌اند. همانطور که حدس می‌زنید، نصب و `import` تمام محتوای سرویس A درون سرویس B بیهوده و خطا خیز است چرا که ما در مهندسی نرم‌افزار علاقه‌مند به «مخفی‌سازی اطلاعات»^۳ هستیم. توضیح دهید چگونه با برقراری DIP می‌توانیم این مشکل وابستگی مستقیم میان A و B را حل کنید.

راهنمایی: یک کلاس AServiceFacadeInterface درون پکیج سومی به نام I قرار دهید - در پایتون که اینترفیس موجود نیست می‌توان از کلاس ABC بهره برد و تمامی توابع این کلاس را نیز `abstractmethod` کرد.

ملاحظه: به قطع برای مشاهده درک شما از راه‌حلتان باید مقداری قطعه کد و مدل UML ارائه دهید.

^۳ Information Hiding

تحقیق ششم: Design By Contract در عمل

پیش‌نیاز:

● Java یا Python یا Javascript

در این تحقیق قصد دارید با Design By Contract آشنا شوید و آن را در عمل به کار بگیرید. در متن تحقیق خود اطمینان حاصل کنید که موارد زیر در نظر گرفته شود:

1. از ابتدا تا دقیقه‌ی ۵۲ از [این ویدیو](#) کلاس درس دکتر رامسین که درباره Design By Contract می‌باشد را

مشاهده کنید. سپس در مقدمه‌ی تحقیق خود بر اساس محتوای این ویدیو و باقی مواردی که در کاوش‌های خود پیدا کرده‌اید متنی مفصل درباره DBC بنویسید و آن را توضیح دهید.

2. تحقیق کنید زبان برنامه‌سازی Eiffel چگونه از این مکانیزم درون خود استفاده کرده است و آن را توضیح دهید. انتظار نمی‌رود در این مورد به دقت زبان Eiffel را یاد بگیرید و مجبور شوید روی آن کدی بزنید. توضیحات کلی‌ای که از اینترنت و یوتیوب (به انضمام نمونه‌کدهایی که از منابع خود پیدا می‌کنید) کفایت خواهد کرد.

3. برای یکی از سه زبان Java یا Python یا Javascript (به انتخاب و راحتی خودتان)، یک کتابخانه پیدا کنید که مکانیزم DBC را به کد بیفزاید و با دانشی که خودتان از آن زبان دارید توضیح دهید:

a. چگونه می‌توان با استفاده از کتابخانه انتخابی، به DBC در زبان مقصد دست یافت.

b. توضیح کلی که چگونه این کتابخانه DBC را به زبان مقصد افزوده است.

c. تا چه حد این کتابخانه توانسته به تمام اصول DBC پایبند باشد.

برای مشاهده‌ی لیستی از کتابخانه‌های موجود در زبان‌های مختلف که DBC را به آن زبان اضافه می‌کند، [این بخش](#) از صفحه‌ی ویکی‌پدیا را مطالعه کنید و یک کتابخانه برای زبان مورد نظرتان را انتخاب کنید. ترجیحا کتابخانه‌ای را انتخاب کنید که معروف‌تر باشد (برای مثال Star بیشتری در گیت‌هاب داشته باشد) و روی Framework دیگری سوار نشده باشد - برای مثال در مورد Java کتابخانه‌ای را انتخاب نکنید که پیشنهادش نصب چارچوب Spring برای ایجاد Web باشد.

4. برای مورد سوم، حتما قطعه کدهایی در تحقیق خود قرار دهید و با توضیح آن قطعه کدها نحوه عملکرد کتابخانه انتخابی را توضیح دهید. قطعه کدهایی که در تحقیق قرار می‌دهید به ترتیب عملیات‌هایی ساده تا اندکی پیچیده را به تصویر بکشند و نشان دهید در هر عملیات و منطق، چگونه DBC خوانایی و عملکرد صحیح کد را تضمین کرده است.

تحقیق هفتم: State Management در Vue

با تشکر از سپهر کیانیان عزیز بابت ارائه این موضوع
پیش‌نیا:

● Vue.js

یکی از مسائل معمول در ایجاد مولفه‌های front-end در سامانه‌ها، مسئله‌ی مدیریت حالت است.

1. ابتدا [این متن](#) را مطالعه کرده و راجع به موارد زیر توضیح دهید.
 - a. مسئله‌ی مدیریت حالت چیست؟
 - b. این مسئله چه مشکلاتی می‌تواند به وجود آورد؟
 - c. رویکردهای معمولی که برای حل این مسئله استفاده می‌شود چیست؟
2. عموماً سر منشا الگوهای حل این مسئله، الگوی طراحی معروف «مشاهده‌کننده»⁴ برگرفته از کتاب معروف GoF است.⁵ در مورد سه وجه این الگو (زمینه، مسئله و راه‌حل) تحقیق کنید سپس توضیح مفصلی درباره این الگو ارائه دهید.
3. پیش از مطالعه بخش‌های چهار و پنج از این تحقیق، راه‌حلی اولیه و نه خیلی مفصلی که با استفاده از الگوی مشاهده‌کننده به‌نظرتان می‌رسد، برای مسئله مطرح شده در بخش ۱، مطرح کنید.
4. یک الگوی بسیار رایج برای حل این مسئله، که عموماً کتابخانه‌ها از آن پیروی می‌کنند، را می‌توانید از [اینجا](#) مطالعه کنید. پس از مطالعه، برای موارد زیر توضیح کافی ارائه دهید.
 - a. این الگو از چه اصولی پیروی می‌کند
 - b. این الگو از چه مولفه‌هایی تشکیل شده است.
5. [Vuex](#) یکی از کتابخانه‌هایی است که در چارچوب⁶ Vue به حل این مسئله می‌پردازد. این کتابخانه همچنین یک الگوی منحصر به فردی برای حل این مسئله ارائه می‌دهد که می‌تواند تمامی قسمت‌های مولفه‌ی front-end را تحت الشعاع قرار دهد. در مورد ساختار کلی این الگو و همچنین قسمت‌های مختلفی که در آن وجود دارد توضیح کافی ارائه دهید.

⁴ Observer

⁵ <https://refactoring.guru/design-patterns/observer>

⁶ Framework

تحقیق هشتم: تحقیق واسط⁷ در عمل

پیش‌نیاز:

● شی‌گرایی در حد Java

در این تحقیق استفاده از مفهوم بسیار مفید واسط را در زبان‌های برنامه‌نویسی مختلف بررسی می‌کنیم.

1. واسط در زبان‌های شی‌گرا چیست و چه تفاوت‌هایی با کلاس انتزاعی⁸ دارد؟ در توضیحات خود به موارد زیر توجه ویژه‌ای داشته باشید.

a. رابطه‌ی تحقق⁹ (بین کلاس و واسط) چه تفاوتی با رابطه توارث¹⁰ (بین دو کلاس) دارد؟ در توضیحات خود به موارد زیر توجه داشته باشید:

i. در زبان جاوا در مورد توارث و در مورد تحقق چه کلیدواژه‌هایی استفاده می‌شود؟

ii. تفاوت شکل این دو رابطه در نمودار کلاسی UML به چه صورت است؟

b. در مورد کلاس انتزاعی حتما باید اصل LSP از اصول SOLID برقرار باشد. در مورد واسط‌ها چطور؟ علت را توضیح دهید.

c. تحقیق کنید چگونه با استفاده از واسط می‌توان به برقراری اصل DIP بیشتر از پیش (که تنها توارث استفاده می‌شد) کمک کرد.

d. فرض کنید کلاس S خدماتی را عهده‌دار است که دو بخش با فاصله زیاد در پایگاه‌کد موسوم به A و B هر کدام نیازمند بخشی از این خدمات هستند. اگر از مفهوم واسط استفاده نشود، A و B کل خدمات S را مشاهده می‌کنند و آن بخش مورد نیاز خود را دریافت می‌کنند. ایراد این موضوع چیست؟ چگونه با استفاده از واسط و برقراری اصل ISP از اصول SOLID این ایراد را می‌توان برطرف کرد؟

2. به دلیل توان برنامه‌نویسی حاصل از استفاده از مفهوم Interface، یک بحثی که همواره در جامعه مهندسی نرم‌افزار وجود دارد، نحوه شبیه‌سازی این مفهوم در زبان‌هایی است که به طور پیش‌فرض این مفهوم را ارائه نمی‌دهند (برای مثال در مورد Java می‌دانیم این مفهوم موجود است).

a. در مورد دو زبان برنامه‌سازی که این مفهوم را به طور پیش‌فرض ندارند تحقیق کنید که به چه شکل می‌توان، واسط را در آن‌ها شبیه‌سازی کرد (پیشنهاد: Python و JS).

b. نبود مفهومی شبیه واسط چگونه مهندسی نرم‌افزار را در این زبان‌ها سخت می‌کند که این بحث در جامعه مهندسی وجود دارد؟

3. در مورد واسط در زبان Go تحقیق کنید:

⁷ Interface Realization

⁸ Abstract

⁹ Realization

¹⁰ Inheritance

- a. آیا رده‌هایی که در این زبان یک واسط را محقق می‌کنند لازم دارند به صورت صریح، تحقق آن واسط را اعلام کنند؟ برای مثال در Java می‌دانید که باید از کلید واژه‌ای که در بخش a.1. ذکر کرده‌اید بر روی کلاس محقق‌کننده‌ی واسط استفاده شود.
- b. تصمیمی که برای کامپایلر go در بخش a گرفته شده چه فوایدی از دید یک مهندس نرم‌افزار خواهد داشت؟