

به نام خدا

آزمون پایان ترم + پاسخ

نیم سال دوم ۱۴۰۱-۱۴۰۲

۳ تیر ۱۴۰۲

سوالات تستی (هر کدام ۱۰ نمره)

۱. منظور از failure در آزمون نرم افزار چیست؟

- a. حضور یک fault در نرم افزار
- b. یک error که در طی اجرای نرم افزار رخ می دهد
- c. یک اشکال در عملکرد یا انحراف از رفتار مورد انتظار نرم افزار
- d. یک defect که در طی فاز تست نرم افزار کشف شده

پاسخ: گزینه ج منبع: اسلاید ۴ چپتر ۲۲

دلیل: مطابق توضیحات اسلاید می باشد.

۲. یکی از ویژگی های آزمون خوب برای یک نرم افزار، Best of Breed بودن است. منظور از این ویژگی برای یک آزمون چیست؟

- a. بین یک دسته از آزمون ها که هدف، منابع زمانی و سخت افزاری مشابهی دارند، آزمونی انتخاب شود که بتواند دسته ای کامل از خطاها را شناسایی کند.
- b. یک آزمون نباید هدفی یکسان با آزمونی دیگر داشته باشد.
- c. آزمون باید با احتمال خوبی بتواند خطاهای آن بخش از نرم افزار را که برای آن نوشته شده است، پیدا کند.
- d. آزمون باید به صورت تخصصی برای یک بخش کوچک از برنامه نوشته شود. تست نباید بزرگ باشد و بخواهد کار چندین آزمون را در خودش انجام دهد و بخش بزرگی از ویژگی های برنامه را پوشش دهد.

پاسخ: گزینه ۱ - ارجاع به صفحه ۳۹۶ کتاب پرسمن ویرایش ۹

۳. درباره انواع روش های آزمون یک نرم افزار، کدام گزینه نادرست است؟

- a. روش آزمون جامع^۱، تمامی مسیرهای ممکن در یک برنامه را پیمایش می‌کند تا بتواند صحت برنامه را در تمامی سناریوهای ممکن بیازماید. با استفاده از پردازنده‌های سریع مخصوص این نوع از آزمون‌ها می‌توان این نوع از آزمون را در زمان معقولی انجام داد.
- b. روش‌های جعبه‌سفید^۲ و جعبه‌سیاه^۳ در آزمون نرم‌افزار، روش‌هایی مادر برای دیگر روش‌ها هستند که تفاوت آن‌ها در دانستن ساختار داخلی برنامه توسط آزمونگر نرم‌افزار است.
- c. روش آزمون بر پایه مدل^۴، فرض می‌کند که به مدل‌های رفتاری که در مراحل ایجاد نرم‌افزار ایجاد شده‌اند، دسترسی دارد.
- d. روش طراحی آزمون مدل-محور^۵، فاصله‌ی روش‌های آزمون جعبه‌سفید و جعبه‌سیاه را کم‌رنگ می‌کند و ریاضیاتی‌ترین و فنی‌ترین روش آزمون نرم‌افزار است.

پاسخ: گزینه ۱

در برنامه‌های پیچیده، هیچ پردازنده‌ای وجود ندارد که بتواند تمامی حالات که تعداد آن‌ها نمایی است را بررسی کند. ارجاع شود به صفحه ۳۸۰ کتاب پرسمن ویرایش ۹

درستی باقی گزینه‌ها در صفحات ۲۲ تا ۲۶ اسلایدهای ۲۳-۲۴-۲۵

4. کدام گزینه در مورد Regression Testing درست است؟

- a. به ما کمک می‌کند تا مطمئن شویم تا قسمت‌هایی از نرم‌افزار که بهشان دست نزده‌ایم، خراب نشده‌اند.
- b. فقط در حین فرآیند User Acceptance Test قابل اجرا است.
- c. همواره به صورت خودکار انجام می‌شود.
- d. پس از commit شدن تغییری روی مخزن کد، Regression Test انجام می‌شود.

پاسخ: گزینه الف منبع: اسلاید ۱۶ چپتر ۲۲

دلیل: گزینه‌ی الف مطابق با تعریف regression testing است؛ regression test این را چک می‌کند که تغییرات یک قسمت روی قسمت‌های دیگر که به آن دست نزده‌ایم، اثر منفی نداشته باشد. در مورد گزینه‌ی چهار، Regression Test پیش از commit شدن تغییرات روی مخزن کد انجام می‌شود تا در صورتی که مشکلی در تغییرات صورت گرفته وجود داشت، کدهای موجود در مخزن کثیف نشوند.

5. کدام یک از گزینه‌های زیر، جزء فواید روش افراز فضای ورودی^۶ برای آزمون یک method محسوب نمی‌شود؟

- a. می‌توان این روش را بدون خودکارسازی اعمال کرد.

¹ Exhaustive Testing

² White-box Testing

³ Black-box Testing

⁴ Model-based

⁵ Model-driven Test Design

⁶ Input-space Partitioning

b. این روش می‌تواند در مراحل مختلف آزمون از جمله آزمون سیستم⁷، آزمون ادغام⁸ و ... اعمال شود.

c. برای استفاده از این روش و فهمیدن فضای ورودی، نیاز به دانش حداقلی نسبت به پیاده‌سازی method است.

d. این روش منطبق با نیاز آزمونگر، می‌تواند تعداد کم یا زیاد آزمون تولید کند.

پاسخ: گزینه ۳ - در این روش، نیازی به دانش پیاده‌سازی نیست و یکی از روش‌های آن که بر پایه interface است، این دلالت بر این موضوع دارد. (ارجاع به صفحه ۸۷ اسلایدهای سری ۲۲-۲۳-۲۴)

6. کدام گزینه در مورد Beta Testing صحیح می‌باشد؟

a. روشی ارزان برای اجرای یک برنامه تضمین کیفیت نرم‌افزار (software quality assurance) می‌باشد.

b. روشی معقول برای آزمودن نرم‌افزار مستقرشده روی زیرمجموعه‌ای از کاربران نهایی است.

c. بهترین راه برای دریافت بازخورد متمرکز کاربر برای پشتیبانی از بهبود رابط کاربری نرم‌افزار می‌باشد.

d. راهی تضمین شده برای به دست آوردن تبلیغات مثبت و شهرت برای محصول می‌باشد.

پاسخ: گزینه ب منبع: اسلاید ۱۷ چپتر ۲۲
دلیل: مطابق با تعریف beta testing می‌باشد.

7. کدام گزینه در مورد Integration Testing که در سطح کوچک انجام می‌شود، صحیح می‌باشد؟

a. کامپوننت‌های تکی⁹ را تست می‌کند.

b. فقط از کامپوننت‌های سیستم مستقرشده در محیط کاربر را می‌آزماید.

c. تعامل بین ماژول‌ها یا زیرسیستم‌های تکی را آزمایش می‌کند.

d. interface هایی را که هر سیستم در اختیار سیستم‌های دیگر می‌گذارد، آزمایش می‌کند.

پاسخ: ج منبع: اسلاید ۱۵ چپتر ۲۲

دلیل: منظور از سطح کوچک، همین موضوع است که تعاملات بین ماژول‌ها یا زیرسیستم‌های تکی را آزمایش می‌کند. Integration Testing می‌تواند در سطوح مختلفی انجام شود؛ مثلاً گزینه «د» در سطح بزرگ انجام می‌شود. Integration Testing در سطح بزرگ، تعاملات بین اجزای کل سیستم یا سیستم‌ها را در برمی‌گیرد که شامل interface هایی است که در اختیار سایر سیستم‌ها یا سرویس‌های بیرونی قرار می‌دهند.

⁷ System Testing

⁸ Integration Testing

⁹ Individual

8. رشید در حال اجرای یک تست است که با نیاز کاربر برای پر کردن یک فیلد خاص با استفاده از یک Drop Down Box که لیستی از مقادیر دارد، مطابقت دارد. در این زمان کدام یک از انواع تست در حال اجرا شدن است؟

- a. White-box Testing
- b. Black-box Testing
- c. Load Testing
- d. Regression Testing

پاسخ: گزینه ب منبع: اسلاید ۲۳ چپتر ۲۳

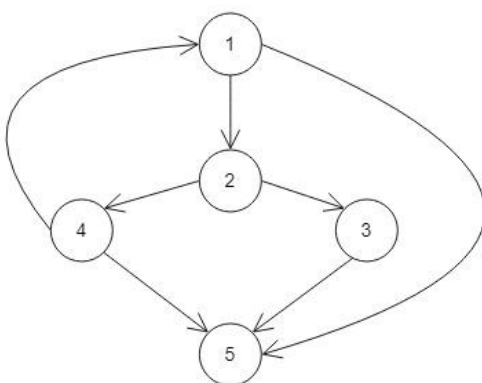
دلیل: مطابق با تعریف black-box testing است. در این حالت به کد کاری نداریم و از آن خبری نداریم و صرفاً می‌خواهیم یک سری ورودی بدهیم و با خروجی مورد انتظارمان چک کنیم. که در اینجا ورودی همان انتخاب کردن از لیست موجود و خروجی پر شدن آن فیلد خاص با مقدار انتخاب شده می‌باشد.

9. هنگامی که فرآیند Verification and Validation یک نرم‌افزار ایجاد شد...

- a. دیگر هرگز تغییر نمی‌کند.
- b. می‌توان آن را تا زمان شروع قرارداد تغییر داد.
- c. می‌توان آن را تا زمان شروع ایجاد (Development) واقعی نرم‌افزار تغییر داد.
- d. می‌توان آن را در طول چرخه حیات نرم‌افزار تغییر داد.

پاسخ: گزینه «د» منبع: اسلاید ۷ چپتر ۲۲

دلیل: با توجه به موارد مختلفی از جمله نیازمندی‌های جدیدی که در طول زمان ممکن است به وجود بیایند، شناسایی شدن ریسک‌های جدید یا تغییر در اهمیت ریسک‌ها، ... فرآیند V&V در طول چرخه حیات می‌تواند تغییر کند.



10. تصویر زیر گراف کنترل حالات یک برنامه‌ی فرضی است

(۱ حالت آغازین و ۵ حالت پایانی است)؛ هر کدام از

گزینه‌های زیر دو مسیر را مشخص کرده است. در کدام

گزینه، دو مسیر مستقل از هم وجود ندارد؟

- a. (5, 1) و (5, 4, 2, 1)
- b. (5, 3, 2, 1) و (5, 4, 2, 1)
- c. (5, 3, 2, 1) و (5, 4, 2, 1, 4, 2, 1)
- d. (5, 4, 2, 1) و (5, 4, 2, 1, 4, 2, 1)

پاسخ: گزینه ۴

دو مسیری که در گزینه‌ی چهار وجود دارند، در یال‌های 1, 2 و 4, 5 اشتراک دارند. البته نمره‌ی این سوال به همه داده می‌شود؛ چرا که فعل به جای «ندارد»، «دارد» بود.

سوالات تشریحی

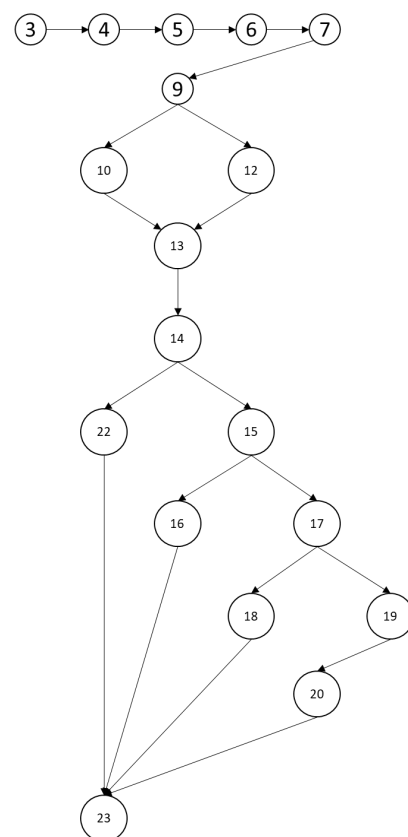
سوال اول: گراف جریان (۲۰ + ۱۰ + ۱۰ نمره)

```
1  int a, b, c
2  bool isTriangle
3  Output(enter a, b and c integers)
4  Input(a, b, c)
5  Output(side 1 is {a})
6  Output(side 2 is {b})
7  Output(side 3 is {c})
8
9  if (a<b+c) and (b<a+c) and (c<b+a)
10     isTriangle = True
11 else
12     isTriangle = False
13
14 if isTriangle
15     if (a == b) and (b == c)
16         Output("equilateral")
17     esle if (a != b) and (a != c) and (b != c)
18         Output("scalene")
19     else
20         Output(isosceles)
21 else
22     Output("not a triangle")
```

با توجه به تکه‌کد زیر به سوالات پاسخ دهید.

الف) گراف جریان مربوط به این تکه‌کد را رسم کنید. برای سادگی، می‌توانید هر خط را معادل یک راس گراف در نظر بگیرید.

پاسخ:



ب) یک مجموعه‌ای پایه از مسیرهای خطی مستقل پیدا کنید.

3, 4, 5, 6, 7, 9, 12, 13, 14, 22, 23

3, 4, 5, 6, 7, 9, 10, 13, 14, 15, 16, 23

3, 4, 5, 6, 7, 9, 10, 13, 14, 15, 17, 18, 23

3, 4, 5, 6, 7, 9, 10, 13, 14, 15, 17, 19, 20, 23

پ) مجموعه‌ای کمینه از موارد آزمون را پیدا کنید که تمام مسیرهای قسمت قبل را پوشش دهد.

{a=1, b=2, c=100}, {a=5, b=5, c=5}, {a=3, b=4, c=5}, {a=7, b=7, c=6}

سوال دوم: آزمودن متد (۲۰ + ۱۰ + ۱۰ + ۱۰ نمره)

متدی داریم که سه عدد صحیح ورودی می‌گیرد و به ما نوع مثلثی را که می‌توانیم با آن بسازیم، خروجی می‌دهد. در این مسئله فرض کنید ما در عالم تنها سه نوع مثلث مختلف الاضلاع، متساوی‌الساقین و متساوی‌الاضلاع را داریم!

1. با استفاده از یک یا چند خصوصیت دلخواه، ورودی‌های این متد را به دسته‌های مختلف افزار کنید.

باید توجه شود که افراز باید تمامی حالات مسئله را در برگرد و هیچ‌کدام نباید همپوشانی ورودی داشته باشند. یک افراز به صورت زیر می‌تواند باشد:

Partition	B_1	B_2	B_3	B_4
$q_1 = \text{"Geometric classification"}$	Scalene	Isosceles	Equilateral	Invalid

2. خصوصیت یا خصوصیت‌هایی را که در بخش قبل استفاده کردید، مبتنی بر عملکرد هستند یا مبتنی بر رابط؟

تنها از یک خصوصیت برای افزار ورودی‌های متد استفاده کردیم (نوع مثلثی که شکل می‌دهند) که این خصوصیت مبتنی بر عملکرد است.

3. نیازمندی‌های آزمون برای پوشش All Combination به دست بیاورید.

از آنجایی که تنها از یک خصوصیت برای افراز ورودی‌های متد استفاده کردیم، پس پوشش All Combination شامل داشتن یک نمونه از هر کدام از دسته‌های B_1 تا B_4 است:

$$\{B_1, B_2, B_3, B_4\}$$

4. کمترین تعداد مورد آزمون را برای پوشش نیازمندی‌های آزمون مرحله قبل، بنویسید.

Test case	(4,5,6)	(3,3,4)	(3,3,3)	(3,4,8)
-----------	---------	---------	---------	---------

سوال سوم: گزاره‌های منطقی (۴ + ۱۸ + ۸ + ۸ + ۱۲ نمره)

با توجه به predicate های داده شده در هر قسمت به سوالات آن پاسخ دهید:

$$P_1 = (a \wedge b) \vee (a \wedge \neg b)$$

1. clause های P_1 مشخص کنید.

دو clause در این predicate داریم که a و b هستند.

2. هر clause چگونه عبارت P_1 را تعیین می‌کند؟

$$P_a = P_{a=true} \oplus P_{a=false}$$

$$\begin{aligned}
&= ((true \wedge b) \vee (true \wedge \neg b)) \oplus ((false \wedge b) \vee (false \wedge \neg b)) \\
&= (b \vee \neg b) \oplus false \\
&= true \oplus false \\
&= true
\end{aligned}$$

$$\begin{aligned}
P_b &= P_{(a=true)} \oplus P_{b=true} \\
&= ((a \wedge true) \vee (a \wedge false)) \oplus ((a \wedge false) \vee (a \wedge true)) \\
&= (a \vee false) \oplus (false \vee a) \\
&= a \oplus a \\
&= false
\end{aligned}$$

3. درباره‌ی بخش قبل توضیح دهید که جواب به دست آمده درباره دو متغیر a و b چه معنایی دارند؟
a همیشه جواب این predicate را تعیین می‌کند و b هیچگاه جواب را تعیین نمی‌کند؛ به عبارت دیگر b نامربوط (irrelevant) است.

$$\begin{aligned}
P_2 &= \forall a \in \mathbb{Z}, \exists b \in \{-1, 1\} \mid (f(a) \wedge (b == 1)) \vee (\neg f(a) \wedge (b == -1)) \\
f(x) &: (x \bmod 2) == 0
\end{aligned}$$

4. P_2 چه عملیاتی را نشان می‌دهد؟

این گزاره نشان‌دهنده‌ی عبارت $a \in \mathbb{Z} \mid (-1)^a$ است (اگر a زوج بود، $b=1$ و اگر a فرد بود، $b=-1$ می‌شود) را همان خروجی/حاصل کار در نظر گرفته‌ایم)

5. موارد آزمون برای تحقق نیازمندی‌های آزمون حاصل از Clause Coverage و Predicate Coverage در P_2 را بنویسید.

Predicate Coverage:

$$f(a) = true[a = 2], b = 1 \rightarrow (true \wedge true) \vee (false \wedge false) = true$$

$$f(a) = true[a = 4], b = -1 \rightarrow (true \wedge false) \vee (false \wedge false) = false$$

با هر روشی دیگری که به دو جواب برای predicate که یکی false و دیگری true شود برسیم، درست است.

Clause Coverage:

$$f(a): true[1], false[2]$$

$b == 1: true[3], false[4]$

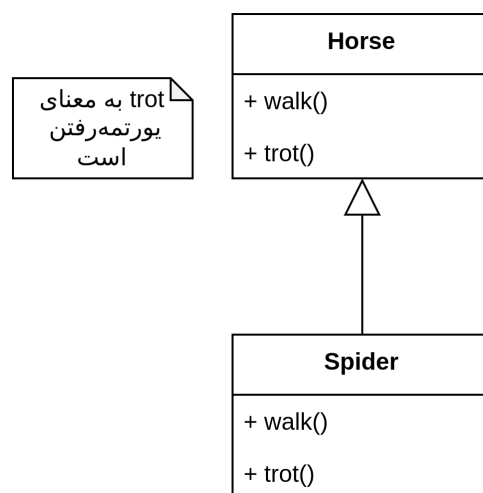
$1, 4 \rightarrow (true \wedge false) \vee (false \wedge true) = false$

$2, 3 \rightarrow (false \wedge true) \vee (true \wedge false) = false$

می‌شد 1 و 3 با هم و 2 و 4 با هم به جای حالت مذکور انتخاب شوند.

سوال چهارم: اصول SOLID (هر مورد ۲۵ نمره)

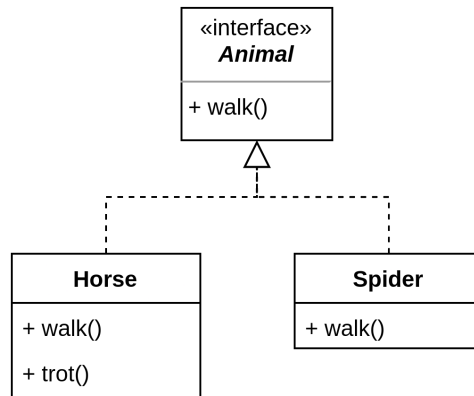
1. در نمودار کلاسی زیر، چه اصلی از SOLID نقض شده است؟ به نظرتان چرا طراح همچین اشتباهی کرده؟ نمودار کلاسی¹⁰ اصلاح‌شده را رسم کنید.



اصل LSP نقض شده است؛ چرا که احتمالاً طراح می‌خواسته در مصرف کد صرفه‌جویی کند (!) و برای عنکبوت تابع walk جدیدی پیاده‌سازی نکند، از همان تابع walk در اسب استفاده کند و آن را به ارث ببرد. اسب باید بتواند یورتمه برود و چون عنکبوت زیرکلاسش است، پس عنکبوت هم باید بتواند یورتمه برود، در حالی که می‌دانیم عنکبوت‌ها یورتمه نمی‌روند؛ پس عنکبوت نمی‌تواند به جای پدرش (=اسب) قرار گیرد و اصل Liskov نقض می‌شود.

راهکار رفع این مشکل این است که هر دوی عنکبوت و اسب، یک فوق‌کلاس مشترک به نام animal داشته باشند و animal تابع walk داشته باشد و اسب و عنکبوت، تابع walk را از animal به ارث ببرند:

¹⁰ Class Diagram



همچنین می‌توان گفت اصل ISP نیز نقض شده است؛ چرا که کلاس عنکبوت مجبور به پیاده‌سازی تابع trot از کلاس اسب شده، در حالی که عنکبوت نیازی به یورتمه‌رفتن ندارد و نباید مجبور به داشتن تابع trot شود.

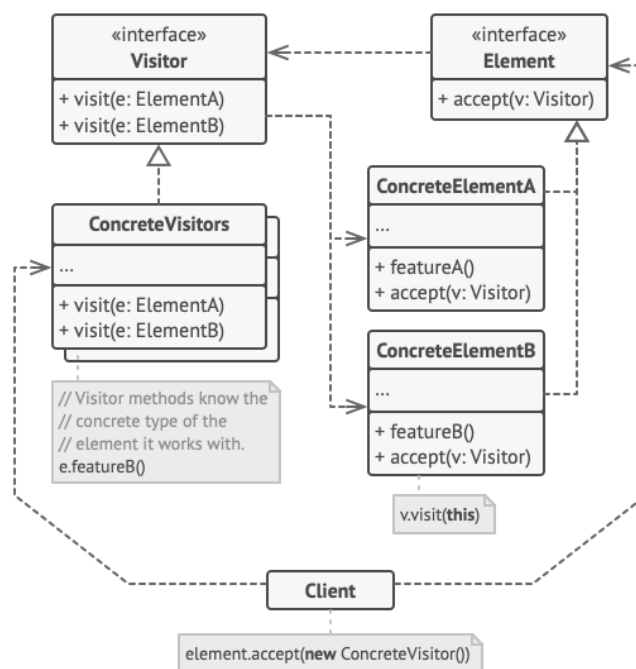
2. تکه‌کد زیر مربوط به سیستم مدیریت حقوق و دست‌مزد یک اداره است. کدام یک از اصول SOLID ممکن است در آن نقض شده باشد؟ توضیح دهید.

```
class Programmer implements Employee {
    function calculatePay() { ... }
    function reportHours() { ... }
    function save() { ... }
}
```

ممکن است اصل SRP نقض شده باشد؛ چرا که به نظر در این اداره، واحدهای مختلفی از توابع موجود در کلاس Employee استفاده می‌کنند؛ مثلاً واحد «محاسبه‌ی دست‌مزد» تابع calculatePay را فراخوانی می‌کند، واحد «منابع انسانی» از تابع reportHours استفاده می‌کند و ... در این صورت، بیش از یک actor باعث تغییر این کلاس می‌شوند (اگر فردا روزی هر کدام از این واحدها تغییری در تابع مربوطه‌شان بخواهند، کلاس Employee باید تغییر کند).

همچنین می‌توان گفت که اصل ISP نیز به خوبی رعایت نشده؛ چرا که برای مثال واحد «محاسبه‌ی دست‌مزد» به تابع calculatePay نیاز دارد ولی به تابع reportHours نیاز ندارد؛ در این حالت واحد محاسبه‌ی دست‌مزد به interface ای وابسته شده که به آن احتیاجی ندارد.

3. الگوی visitor یکی از الگوهای طراحی شی‌گرا است. ساختار این الگو به صورت زیر است:

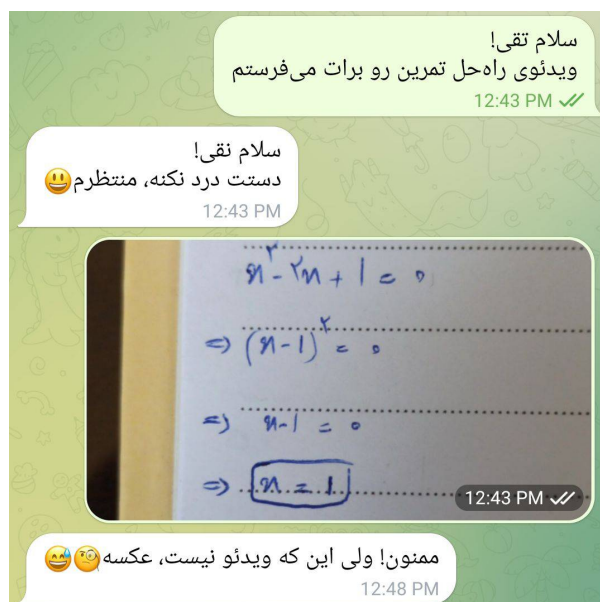


به نظرتان کدام دو اصل SOLID در این الگو رعایت نشده است؟ توضیح دهید.

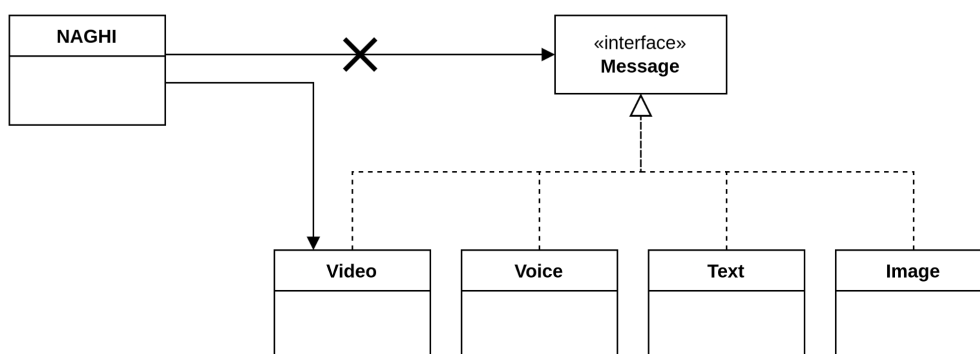
از آنجایی که رابط Visitor به جای این که به رابط Element دید داشته باشد، به صورت مستقیم به کلاس‌های ConcreteElement دید دارد، پس در این ارتباط اصل DIP نقض شده است.

چون اصل DIP نقش شده، پس اصل OCP نیز رعایت نمی‌شود؛ چرا که یکی از لازمه‌های open to extension بودن، وابستگی به رابط سطح بالا به جای وابستگی به کلاس‌های concrete است؛ مثلاً در اینجا اگر بخواهیم یک ConcreteElement جدید اضافه کنیم، باید تمام Visitor ها را تغییر دهیم.

4. قسمتی از چت نقی و تقی را در تصویر زیر مشاهده می‌کنید (سمت راست نقی و سمت چپ تقی است). نقی می‌خواسته از راه‌حل تمرینش برای تقی فیلم بگیرد و بفرستد، ولی کمی بعد به نظرش رسیده که عکس کافی است و نیازی به ارسال ویدئو نیست. به نظرتان پیام اول نقی باید چه‌گونه می‌بود که اگر به جای ویدئو، عکس یا صوت هم می‌فرستاد، باز هم ابهامی برای تقی پیش نمی‌آمد؟ نقی کدام اصل SOLID را رعایت نکرده است؟



نقی در اینجا اصل DIP را رعایت نکرده؛ او در پیام اولش به جای این که به رابط سطح بالای «پیام» اشاره کند (همان رابطه‌ای که در نمودار زیر ضربدر خورده)، به یک پیام concrete (پیام از نوع ویدئو) اشاره کرده و وقتی نظرش عوض شده و پیام از نوع عکس فرستاده، حرف اولش ابهام‌دار شده است.



بهتر بود که نقی در پیام اولش، اسمی از «ویدئو» که زیرکلاسی از رابط «پیام» است، نمی‌آورد تا در ادامه هر چقدر هم که نظرش عوض شد، می‌توانست هر نوع پیامی را که می‌خواهد، بفرستد.

البته می‌توان گفت که نقی با رعایت نکردن DIP، اصل OCP را هم رعایت نکرده؛ چرا که پس از گفتن پیام اول، راه را برای extension بسته و جز پیام ویدئویی، نمی‌تواند پیام دیگری را بدون ابهام بفرستد.

سوال پنجم: معماری میکروسرویس (۵۰ + ۳۰ + ۳۰ نمره)

به عنوان یک معمار نرم‌افزار، روزی باخبر شدید که یک معماری جدید به عنوان معماری میکروسرویس (Microservice) کشف شده است. احتمالاً با ذوق و شوق به مطالعه آن پرداخته‌اید! روزی در کلاس درس مهندسی نرم‌افزار نیز اسلایدی ارائه و در آن الگوهای معماری میکروسرویس توضیح داده شد. شما که با این معماری آشنا بودید، صرفاً مفاهیم آن را مرور و دانش خود را به‌روز کردید.

در این سوال از شما می‌خواهیم که با دانش قبلی و اسلایدی که در کلاس ارائه شد، سیستمی دلخواه با استفاده از معماری میکروسرویس طراحی کنید:

1. ابتدا از شما می‌خواهیم دست به قلم شوید و شکلی کلی از معماری سیستم دلخواه خود را رسم کنید. می‌توانید اجزا مختلف سیستم را با مستطیل ترسیم کرده و با فلش‌هایی به هم متصل کنید. لزومی ندارد که از زبان مدل‌سازی خاصی (مانند UML) استفاده کنید.

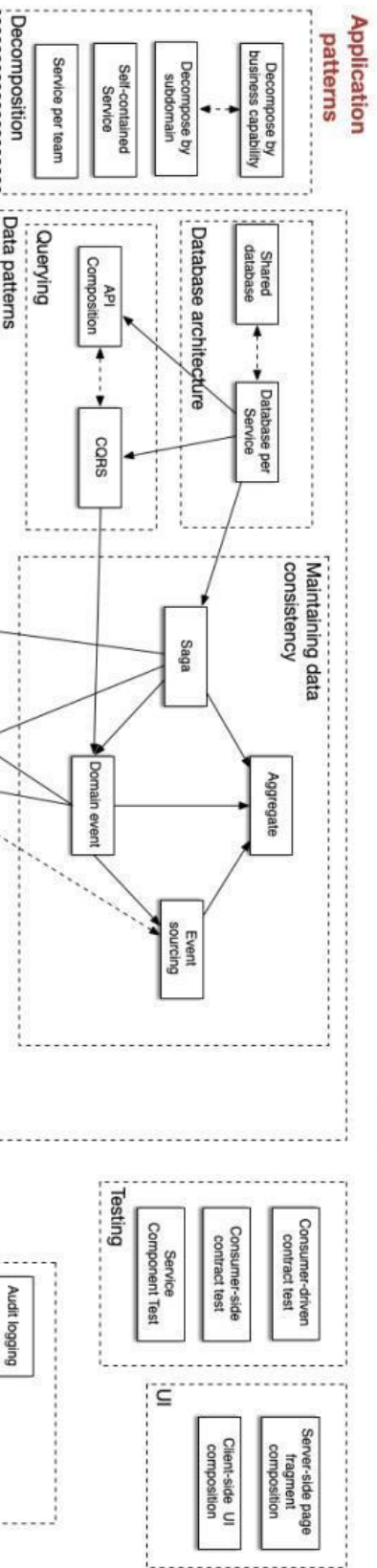
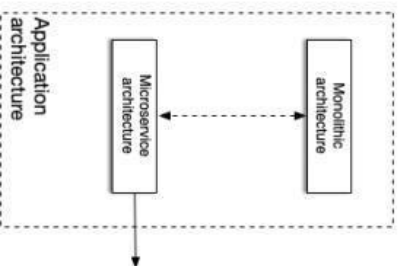
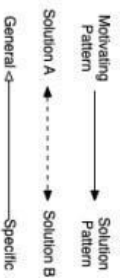
2. در طراحی معماری بالا، شما یک سری «تصمیمات معمارانه» گرفته‌اید. سه مورد از این تصمیمات معمارانه‌ی را توضیح دهید؛ مثلاً در این بخش می‌توانید بنویسید که به فلان دلیل از saga استفاده کردم.

3. در مورد سه تا از مفاهیمی که به صورت کلی در معماری میکروسرویس وجود دارد، توضیح دهید؛ مثلاً می‌توانید بگویید Service per VM به این مفهوم است که
لطفاً مواردی را که در بخش دو نوشته‌اید، در این بخش توضیح ندهید.

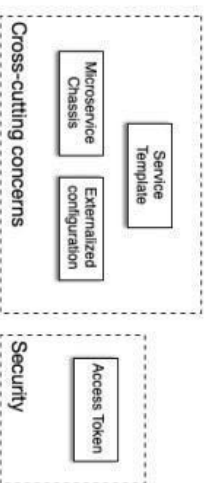
(تصویری از اسلاید مربوط به میکروسرویس که در کلاس ارائه شد، در صفحه‌ی بعد آمده است)

پاسخ بخش‌های ۱ و ۲ این سوال بسته به سیستم دلخواه انتخاب شده دارد.
در بخش ۳ نیز، توضیح هر سه مفهومی که در معماری میکروسرویس وجود دارد، قابل قبول است.

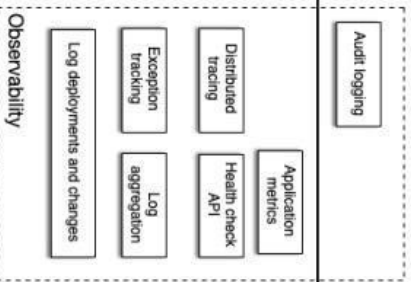
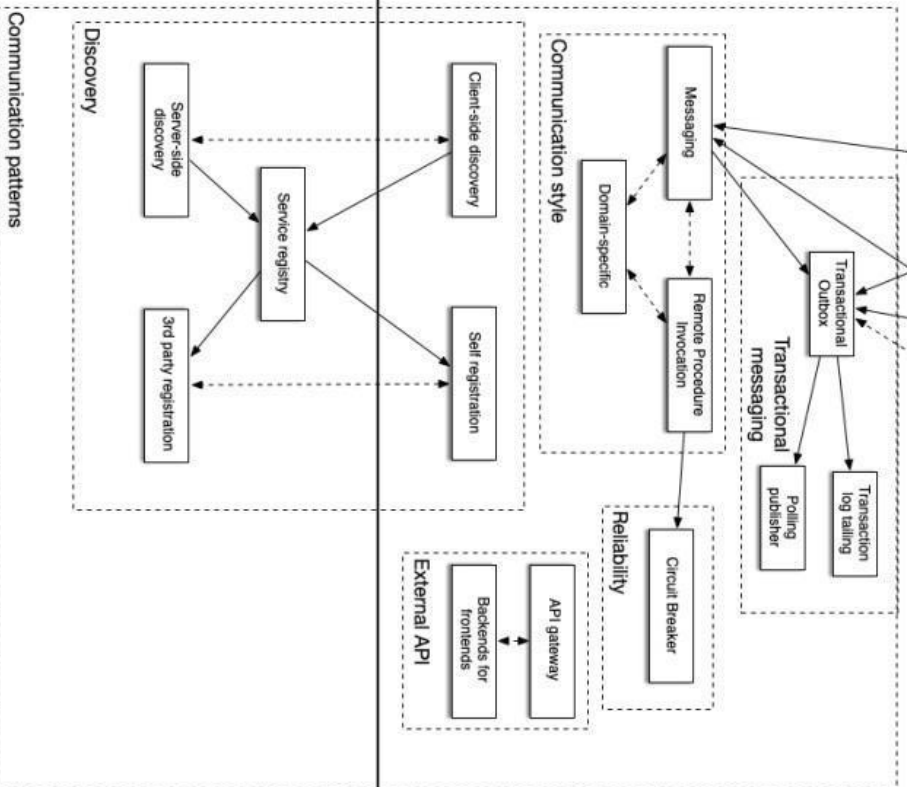
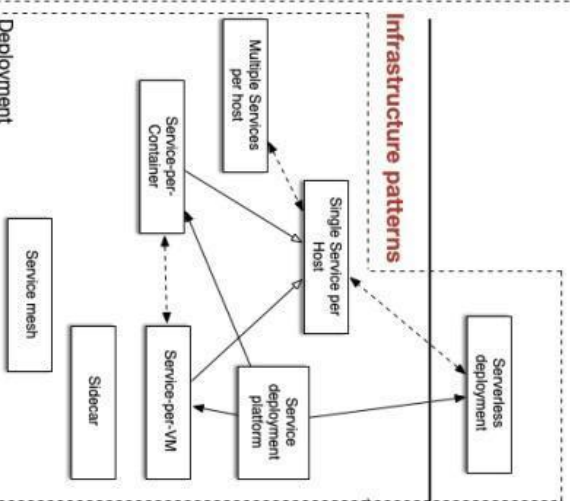
The Microservice Architecture Pattern Language



Application Infrastructure patterns



Infrastructure patterns



سوال ششم: مفاهیم کلی (۲۵ + ۲۵ + ۶۰ + ۴۰ نمره)

کپل، مپل و تپل سه تا از دانشجویان باحال دانشگاه صنعتی شریف هستند. آن‌ها از کودکی دوست داشتند که مهندس کامپیوتر باشند و در این زمینه فعالیت کنند. الان که به انتهای تحصیل خود نزدیک شده‌اند، تصمیم به راه‌اندازی کسب‌وکاری در این حوزه گرفته‌اند.

کپل تخصص خوبی در حوزه‌ی مهندسی نرم‌افزار و به ویژه متدولوژی اسکرام دارد. به‌علاوه، مفاهیم مربوط به نیازمندی‌ها را بسیار خوب درک می‌کند.

مپل طراح و معمار خوبی است و مفاهیم این حوزه را بسیار خوب درک می‌کند. به علاوه مدت‌ها برنامه‌نویسی کرده و با این مفاهیم آشنا است. او دوستی به نام خپل دارد که مفاهیم UI/UX را به خوبی می‌شناسد.

تپل اما، آزمون‌گر خبره‌ای است و با مفاهیم آزمون نرم‌افزار آشنایی زیادی دارد.

امروز کپل، مپل و تپل، برای ایجاد کسب‌وکار جدیدشان جلسه‌ای تشکیل داده‌اند. خپل هم به صورت آنلاین در جلسه حاضر شده (چرا که خواب ماند!). لطفاً در خصوص این جلسه به موارد زیر پاسخ دهید:

1. اگر این یک جلسه ایده‌آل بوده باشد، چگونه برگزار شده است؟ (هدف از این سوال بررسی جلسات اصولی است)
2. به نظر شما چه کسب و کاری پیشنهاد شده است؟ چرا؟ (هدف، بررسی نگاه شما در خصوص کسب‌وکارهاست)
3. دو مورد از پیشنهاداتی را که هر کدام از اعضا مطرح کرده‌اند، بنویسید. (در مجموع ۸ مورد) (هدف از این سوال، بررسی شناخت هر یک از حوزه‌هاست)
4. لطفاً به هریک از اعضا جلسه یک پیشنهاد برای بهترشدن در زمینه‌ی تخصصی‌اش بدهید. (هدف، شناخت رویکرد بهبود در ذهن شماست)

هدف این سوال سنجش تسلط شما بر مفاهیم کلی آموخته در درس مهندسی نرم‌افزار است، از این رو این پاسخی مشخص ندارد. نمره‌دهی بر اساس میزان پختگی و عمق جواب‌هایتان انجام می‌شود.



“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”

- Martin Fowler