

Numer indeksu:

Egzamin 13 czerwca 2009

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Pytanie 11. Dopasuj cechy dobrej specyfikacji do ich opisu (do każdej litery dopisz właściwą cyfrę):

- a) Poprawność:
- b) Jednoznaczność:
- c) Kompletność:
- d) Spójność:
- e) Możliwość porządkowania:
- f) Weryfikowalność:
- g) Modyfikowalność:
- h) Możliwość śledzenia:

- 1) Czy wymagania posiadają identyfikatory i wynikają bezpośrednio z wymagań wyższego poziomu?
- 2) Czy wymagania zamawiającego są zgodne z wizją i niesprzeczne ze sobą?
- 3) Czy zamieszczono wszystkie wymagania funkcjonalne i pozafunkcjonalne?
- 4) Czy każde wymaganie ma przyporządkowane kryterium jakości, które można przetestować?
- 5) Czy wymagania mają tylko jedną interpretację?
- 6) Czy można w łatwy sposób wprowadzać zmiany do specyfikacji wymagań?
- 7) Czy każde wymaganie ma atrybuty umożliwiające określenie wagi i znaczenia?
- 8) Czy rzeczywiście tak ma działać system?

Pytanie 12. Uporządkuj numerując od 1 do 6 warstwy typowego programu zrealizowanego w modelu wielowarstwowym; 1 to warstwa najwyższa, najbliższa użytkownikowi programu (dopisz numer):

- a) aplikacja (workflow, zmiana stron/okien, stan sesji, kontroler aplikacji) –
- b) infrastruktura biznesowa (niskopoziomowe usługi biznesowe, reużywane w wielu dziedzinach) –
- c) dziedzina (usługi biznesowe, reguły dziedziny, serwisy) –
- d) usługi techniczne (bezpieczeństwo, utrwalanie) –
- e) prezentacja (interfejs użytkownika, okienka UI, raporty) –
- f) podstawy (usługi podstawowe, struktury danych, obsługa wątków / baz danych / sieci) –

Pytanie 13. Prototyp interfejsu użytkownika to artefakt z fazy (wybierz jedno):

- a) zbierania wymagań
- b) analizy
- c) projektowania
- d) implementacji
- e) testów

Pytanie 14. Dostajesz do przetestowania program obliczający znak wyniku mnożenia dwóch liczb (odpowiedzi programu: ujemny, zero, dodatni). Nie mając dostępu do źródeł programu masz opracować zbiór testów, wyczerpujących wszystkie istotnie różne przypadki, które program musi uwzględnić. Czy twoje postępowania to (wybierz ile chcesz):

- a) testy czarnej skrzynki
- b) testy białej skrzynki
- c) testy obserwacyjnej równoważności

Pytanie 15. W dobrze prowadzonym projekcie informatycznym dokumenty analityczno-projektowe stworzone na początku projektu (wybierz jedno):

- a) nie powinny być zmieniane do końca projektu
- b) powinny być aktualizowane po zakończeniu każdej fazy projektu
- c) powinny być aktualizowane co 3-5 miesięcy (zależnie od wielkości projektu)
- d) powinny być aktualizowane w razie każdej znaczącej zmiany w wymaganiach

Pytanie 16. Dwie główne techniki podnoszące skuteczność projektowaniu i modelowaniu to (wybierz jedno):

- a) Dekompozycja i agregacja
- b) Abstrakcja i dekompozycja
- c) Abstrakcja i generalizacja
- d) Generalizacja i agregacja
- e) Żadne z powyższych

Pytanie 17. Które zdanie najpełniej oddaje cele zarządzania konfiguracją (wybierz jedno):

- a) zapewnienie narzędzi przyspieszających konfigurowanie skomplikowanych systemów informatycznych
- b) zapewnienie kontrolowania wersji i rejestrowania wprowadzanych zmian
- c) umożliwienie śledzenia związków między zmianami a elementami konfiguracji poprzez zapewnienie kontroli wersji i rejestru wprowadzanych zmian

Pytanie 18. Inżynieria oprogramowania (wybierz jedno):

- a) to nauka ścisła, której stosowanie gwarantuje sukcesy w realizacji projektów informatycznych
- b) to dyscyplina empiryczna, która daje narzędzia zwiększające szanse powodzenia projektów informatycznych
- c) to zbiór niezwiązanych praktyk, które nie mają wpływu na powodzenie projektów, ale robią dobre wrażenie na klientach

Pytanie 19. Stosowanie najlepszych praktyk wytwórczych oprogramowania (wybierz jedno):

- a) gwarantuje produkcję dobrej jakości oprogramowania nawet przez kiepskie zespoły
- b) obniża koszty pracy zespołów złożonych z wybitnej klasy specjalistów
- c) pozwala na poprawę jakości oprogramowania i obniżenie kosztów jego produkcji przez dostatecznie dobre zespoły

Pytanie 20. Dla każdego z podanych typów wymagań niefunkcjonalnych podaj jeden przykład miary, jaką można zastosować do weryfikacji takiego wymagania:

- a) Wydajność:

.....
.....
.....

- b) Niezawodność:

.....
.....
.....

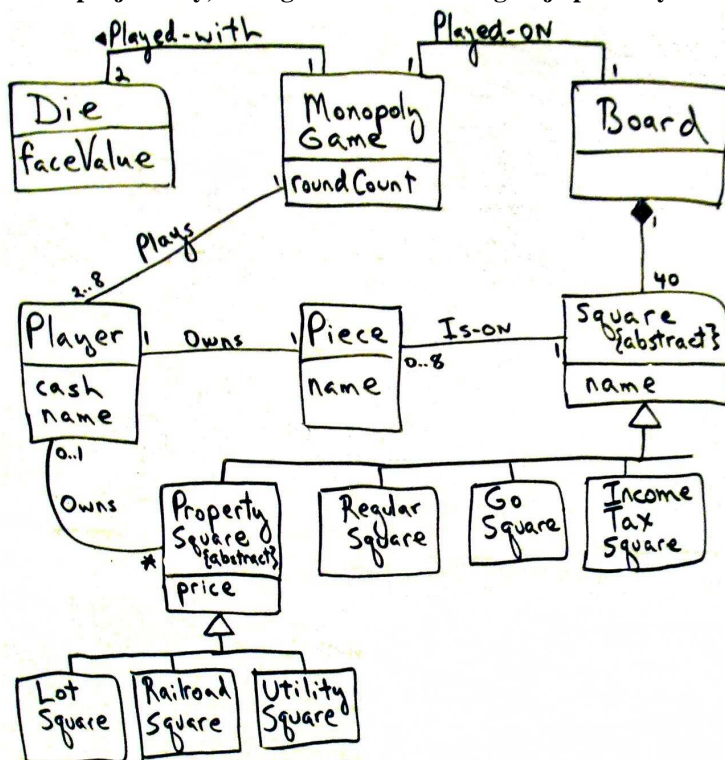
- c) Przenośność:

.....
.....
.....

- d) Skalowalność:

.....
.....
.....

Pytanie 21. Wskaż jeden wzorzec projektowy, którego zastosowanie sugeruje poniższy model klas. Podaj opis wzorca.



.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Pytanie 22. Na które z poniższych pytań powinno dać się łatwo odpowiedzieć po wdrożeniu zarządzania konfiguracją (wybierz ile chcesz):

- a) Jaka platforma jest wymagana dla danej wersji systemu?
- b) Ile kosztowały prace nad daną wersją?
- c) Jaki jest poziom zadowolenia użytkowników danej wersji?
- d) Które wersje systemu są zależne od zmiany danego komponentu?
- e) Jakie komponenty zmodyfikowano przy realizacji danej zmiany?

Pytanie 23. Wzorce projektowe (wybierz ile chcesz):

- a) To ogólnie znane rozwiązania dla dobrze zbadanych problemów.
- b) Są wszystkie opisane w książce "Wzorce projektowe", której autorami są Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.
- c) Czasami dwa wzorce podpowiadają sprzeczne rozwiązania.
- d) Posiadają proste nazwy, które ułatwiają komunikację w zespole.

Pytanie 24. Uzupełnij algorytm zarządzania zmianą (do każdej litery przypisz liczbę):

przyjęcie wniosku zmiany

a) _____

jeżeli (wniosek zasadny) wtedy

b) _____

c) _____

d) _____

jeżeli (proponycja przyjęta) wtedy

repeat

e) _____

f) _____

until (jakość wyniku adekwatna)

g) _____

w przeciwnym przypadku

h) _____

w przeciwnym przypadku

odrzućcie wniosku zmiany

1. analiza sposobu implementacji
2. analiza wpływu na harmonogram i kosztorys
3. analiza wniosku zmiany
4. odrzucenie wniosku zmiany
5. przedstawienie wyniku do weryfikacji
6. przygotowanie propozycji zmiany
7. utworzenie nowego wydania systemu
8. wprowadzenie zmiany

Pytanie 25. Podaj przykład krótkiej funkcji (deklaracja i definicja) dla której wybór przypadków testowych metodą białej (przezroczystej) skrzynki jest niewystarczający.

.....
.....
.....
.....
.....

Pytanie 26. Wskaż elementy służące zapewnianiu jakości oprogramowania (wybierz ile chcesz):

- a) Architektura systemu opiera się na łatwych do rozpoznania wzorcach projektowych
- b) Dane, architektura i interfejsy są opisywane oddzielnie
- c) Projekt interfejsów ogranicza złożoność powiązań modułów ze środowiskiem zewnętrznym i innymi modułami
- d) Projekt jest tworzony sprawdzonymi, powtarzalnymi metodami

Pytanie 27. Czym się różnią testy komponentów (unit tests) od testów integracyjnych:

.....
.....
.....
.....
.....

Pytanie 28. Co w testach przezroczystej skrzynki oznacza kompletność ścieżek:

.....
.....
.....
.....
.....

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Zadanie	Czas trwania	Zależności
Z1	2	
Z2	5	Z1
Z3	7	Z1
Z4	2	Z1
Z5	4	Z4
Z6	3	Z2, Z3
Z7	9	Z5
Z8	2	Z7
Z9	8	Z6
Z10	6	Z9
Z11	10	Z5, Z6, Z10
Z12	12	Z8, Z11
Z13	8	Z9
Z14	5	Z13
Z15	5	Z12, Z14

Pytanie 31. Wskaż wzorce, wykorzystujące zasadę Kompozyt (ang. Composite) (wybierz wiele):

- a) Strategia (ang. Strategy)
- b) Komenda (ang. Command)
- c) Fabryka (ang. Factory)
- d) Singleton (ang. Singleton)
- e) Adapter (ang. Adapter)

Pytanie 32. Główny scenariusz przypadku użycia systemu (wybierz wiele):

- a) nie uwzględnia negatywnego przebiegu (w sensie określonej przez niego funkcjonalności)
- b) musi być zawarty w każdej postaci opisu przypadków użycia (zwięzłej, zwykłej i pełnej)
- c) opisuje proces biznesowy uwzględniający wykorzystanie systemu w być może niespójnym przedziale czasowym (np. na przekroju kilku dni)

Pytanie 33. Wybierz maksymalny pod względem liczby zaznaczonych podpunktów zestaw praktyk/procesów wytwórczych, które można stosować (w całości, bądź częściowo) w ramach jednego projektu (wybierz wiele):

- a) Unit Tests
- b) Rational Unified Process
- c) Agile Unified Process
- d) Extreme Programming

Pytanie 34. Przy rysowaniu modelu systemu (np. w postaci diagramów UMLowych) istotne jest (wybierz jedno):

- a) ścisłe przestrzeganie reguł formalnych notacji i unikanie mieszania rodzajów notacji (np. UML z ERD)
- b) zapewnienie, że tworzone diagramy będą czytelne i ułatwią lekturę dokumentacji, której towarzyszą
- c) rysowanie diagramów we właściwej kolejności – najpierw diagramy statyczne, potem dynamiczne

Pytanie 35. Użycie punktów funkcyjnych do mierzenia pracochołności zadania programistycznego:

- a) Umożliwia porównywanie produktywności zespołów korzystających z różnych języków programowania
- b) Nie jest zalecane dla systemów przetwarzania danych, w których dominują operacje wejścia wyjścia
- c) Jest możliwe jedynie dla przedsięwzięć prowadzonych przy pomocy języków 4GL

Pytanie 36. Pisząc dokument „Wizja” należy (wybierz ile chcesz):

- a) znać listę tabel w tworzonej bazie danych
- b) znać teściową kierownika finansowego projektu
- c) stworzyć listę komponentów systemu w rozbiciu na warstwy (przynajmniej dwie)
- d) znać główną przyczynę, motywację stojącą za pomysłem stworzenia produktu
- e) sformułować problem, który produkt ma rozwiązać
- f) ustalić zakres projektu, jego granice
- g) opisać w formie przypadków użycia komunikację pomiędzy komponentami systemu
- h) opisać wszystkie ważne cechy, funkcje produktu, który ma powstać

Pytanie 37. Które z podanych perspektyw (wymiarów) projektowania są wspierane przez UML (wybierz ile chcesz):

- | | |
|--|---|
| a) Identyfikacja i klasyfikacja obiektów | g) Szczegółowe specyfikacje funkcji |
| b) Związki ogół-szczegół i całość-część | h) Dekompozycja od ogółu do szczegółu |
| d) Opisy atrybutów obiektów | i) Opis pełnych ścieżek przetwarzania danych |
| e) Dzielenie modeli na fragmenty | j) Komunikacja między obiektami za pomocą komunikatów lub zdarzeń |
| f) Stany i przejścia między stanami | |

Pytanie 38. Istotą programowania ekstremalnego jest (wybierz ile chcesz):

- a) ścisła współpraca z klientem
- b) opracowanie testów, przed kodowaniem testowanego kodu
- c) nierealistyczne skracanie harmonogramów
- d) nieprzestrzeganie zasad BHP
- e) praca w systemie 3 zmianowym dzięki wykorzystaniem outsourcingu i offshoringu w krajach położonych w innej strefie czasowej (najlepiej zbliżone do: 8:00-16:00 specyfikowanie; 16:00-24:00 kodowanie; 0:00-8:00 testowanie)

Pytanie 39. W ciągu dwóch tygodni powinna skończyć się półroczna praca nad powyższym systemem, niestety nie wszystkie cele udało się osiągnąć. W szczególności prace nad interfejsem graficznym umożliwiającym rejestrowanie zdarzeń dotyczących kurdła nie zostały zakończone. W tej sytuacji sensownymi działaniami są (wybierz ile chcesz):

- a) dołączenie pięciu nowych programistów do istniejącego zespołu czterech programistów
- b) wydłużenie czasu pracy do 10 godzin dziennie
- c) nakazanie testerom zgłaszania tylko najpoważniejszych usterek (tak by nie rozpraszać uwagi programistów mniej istotnymi błędami)
- d) wydłużenie czasu realizacji projektu (dostarczenie oprogramowania o pełnej funkcjonalności w późniejszym terminie)