

**ΑΝΩΤΑΤΟ ΣΥΜΒΟΥΛΙΟ ΕΠΙΛΟΓΗΣ ΠΡΟΣΩΠΙΚΟΥ  
ΚΕΝΤΡΙΚΗ ΕΠΙΤΡΟΠΗ ΔΙΑΓΩΝΙΣΜΟΥ**

ΠΡΟΚΗΡΥΞΗ 1Γ/2022

ΟΜΑΔΑ Δ

Κλάδοι ΠΕ-ΤΕ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εξέταση στο μάθημα **Σχεδιασμός και Ανάπτυξη Διαδικτυακών Εφαρμογών**  
Σάββατο 22 Οκτωβρίου 2022

Το **ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ** που ακολουθεί αφορά στο μάθημα Σχεδιασμός και Ανάπτυξη Διαδικτυακών Εφαρμογών και αποτελείται από εξήντα πέντε (65) ερωτήσεις πολλαπλών επιλογών, **βαθμολογικά ισοδύναμες**, με τέσσερις (4) εναλλακτικές απαντήσεις (Α, Β, Γ, Δ), από τις οποίες **μία (1) μόνο είναι η ορθή**.

Το μάθημα βαθμολογείται με άριστα τις εκατό (**100**) μονάδες, ενώ ως βαθμολογία **βάσης** του μαθήματος ορίζονται οι **πενήντα πέντε (55)** μονάδες. Για κάθε λανθασμένη απάντηση εφαρμόζεται **αρνητική βαθμολόγηση** με συντελεστή μείον 35%.

Για τις απαντήσεις σας να χρησιμοποιήσετε το **Απαντητικό Φύλλο** σύμφωνα με τις οδηγίες που σας έχουν διανεμηθεί.

## ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ

### 1. Το Λογισμικό Ανοικτού Κώδικα (Open Source):

- α) Διανέμεται δωρεάν χωρίς δικαιώματα χρήσης
- β) Δημοσιοποιεί μόνο τον πηγαίο κώδικα
- γ) Διανέμεται μαζί με τον πηγαίο κώδικα και οι χρήστες του έχουν το δικαίωμα να το χρησιμοποιούν, να το τροποποιούν και να διαθέτουν το τροποποιημένο λογισμικό
- δ) Είναι ελεύθερο προς χρήση χωρίς τεχνική υποστήριξη

### 2. Ως κύκλος ζωής λογισμικού ορίζεται:

- α) Το χρονικό διάστημα από την εγκατάσταση του λογισμικού μέχρι τη διακοπή της χρήσης του
- β) Το χρονικό διάστημα από τη σύλληψη της ιδέας ανάπτυξης του λογισμικού μέχρι την απόσυρση του προϊόντος
- γ) Το χρονικό διάστημα λειτουργικής χρήσης κάθε έκδοσης του λογισμικού
- δ) Το χρονικό διάστημα κατά τη διάρκεια του οποίου παράγονται νέες εκδόσεις του λογισμικού

### 3. Το λογισμικό ορίζεται ως:

- α) Το σύνολο προγραμμάτων, η τεκμηρίωση και διάταξη δεδομένων
- β) Το σύνολο προγραμμάτων
- γ) Η τεκμηρίωση και διαμόρφωση των δεδομένων
- δ) Κανένα από τα παραπάνω

**4. Η διαχείριση έργου με τη μεθοδολογία Agile Scrum:**

- α) Αδιαφορεί για τη διάρκεια υλοποίησης του έργου
- β) Δίνει έμφαση στην τήρηση του μακροχρόνιου προγραμματισμού έργου
- γ) Δίνει έμφαση στη μείωση του κόστους
- δ) Δίνει έμφαση στη σταδιακή επαυξητική πρόοδο

**5. Το/α κύριο/α χαρακτηριστικό/ά του καλού λογισμικού είναι η:**

- α) Ανάπτυξη
- β) Συντηρησιμότητα και λειτουργικότητα
- γ) Λειτουργικότητα
- δ) Συντηρησιμότητα

**6. Ποιος πρότεινε το σπειροειδές μοντέλο;**

- α) Barry Boehm
- β) Roger Pressman
- γ) B. Royce
- δ) IBM

**7. Ανεξάρτητα από το πεδίο εφαρμογής, το μέγεθος του έργου ή την πολυπλοκότητα, οι εργασίες ανάπτυξης λογισμικού μπορούν να χωριστούν σε τρεις γενικές φάσεις εργασίας: στη φάση A, η οποία επικεντρώνεται στο τι, στη φάση B, η οποία επικεντρώνεται στο πώς, και στη φάση Γ, η οποία επικεντρώνεται στην αλλαγή.**

**Επιλέξτε τη σωστή αντιστοίχιση των φάσεων (A), (B) και (Γ) με τα παρακάτω:**

- i. υποστήριξης
- ii. σχεδίασης
- iii. προσδιορισμού απαιτήσεων

- α) A = iii, B = ii, Γ = i
- β) A = iii, B = i, Γ = ii
- γ) A = i, B = ii, Γ = iii
- δ) A = ii, B = i, Γ = iii

**8. Ποιες είναι οι αρχές του ευέλικτου μανιφέστου (Agile Manifesto);**

- α) Ικανοποίηση πελατών
- β) Διά ζώσης επικοινωνία μέσα στην ομάδα ανάπτυξης
- γ) Οι αλλαγές στις απαιτήσεις είναι ευπρόσδεκτες
- δ) Όλα τα παραπάνω

**9. Debugging λογισμικού είναι ο/η:**

- α) Προσδιορισμός της εργασίας που πρόκειται να μηχανογραφηθεί
- β) Δημιουργία κώδικα προγράμματος
- γ) Δημιουργία του αλγορίθμου
- δ) Εύρεση και διόρθωση σφαλμάτων στον κώδικα του προγράμματος

**10. Επιλέξτε την εσφαλμένη πρόταση όσον αφορά τα Αντικείμενα (Objects).**

- α) Τα αντικείμενα είναι αφαιρέσεις του πραγματικού κόσμου
- β) Τα αντικείμενα δεν μπορούν να διαχειριστούν τους εαυτούς τους
- γ) Τα αντικείμενα ενθυλακώνουν την κατάσταση τους
- δ) Όλα τα παραπάνω

**11. Τι ενθυλακώνει (encapsulates) τόσο δεδομένα όσο και λειτουργίες επί των δεδομένων;**

- α) Object
- β) Class
- γ) Super Class
- δ) Sub Class

**12. Ο / Η \_\_\_\_\_ είναι ένας μηχανισμός που επιτρέπει σε πολλά αντικείμενα σε μια ιεραρχία κλάσεων να έχουν διαφορετικές μεθόδους με την ίδια υπογραφή (signature)**

- α) Συσσωμάτωση (Aggregation)
- β) Πολυμορφισμός (Polymorphism)
- γ) Κληρονομικότητα (Inheritance)
- δ) Όλα τα παραπάνω

**13. Η γενίκευση (generalization) στις αντικειμενοστρεφείς γλώσσες προγραμματισμού εφαρμόζεται με:**

- α) Κληρονομικότητα (Inheritance)
- β) Πολυμορφισμό (Polymorphism)
- γ) Ενθυλάκωση (Encapsulation)
- δ) Αφηρημένες κλάσεις (Abstract Classes)

**14. Ποιο(α) από τα παρακάτω αποτελεί πλεονέκτημα της αντικειμενοστρεφούς σχεδίασης (OOD);**

- α) Ευκολότερη συντήρηση
- β) Τα αντικείμενα μπορούν να γίνουν κατανοητά ως αυτόνομες οντότητες
- γ) Τα αντικείμενα είναι δυνητικά επαναχρησιμοποιήσιμα συστατικά (components)
- δ) Όλα τα παραπάνω

**15. Τα αντικείμενα (Objects) εκτελούνται**

- α) Ακολουθιακά
- β) Παράλληλα
- γ) Ακολουθιακά και παράλληλα
- δ) Κανένα από τα παραπάνω

**16. Οι Grady Booch, James Rumbaugh και Ivar Jacobson συνδύασαν τα καλύτερα χαρακτηριστικά της αντικειμενοστρεφούς ανάλυσης, που ο καθένας είχε καθιερώσει, σε μια νέα μέθοδο αντικειμενοστρεφούς σχεδίασης γνωστή ως:**

- α) HTML
- β) XML
- γ) UML

δ) SGML

**17. Ποια η διαφορά μεταξύ των επιχειρησιακών απαιτήσεων (business requirements) και των απαιτήσεων χρηστών (user requirements);**

- α) Οι επιχειρησιακές συνδέονται με τους επιχειρησιακούς στόχους του οργανισμού για τον οποίο αναπτύσσεται το λογισμικό, ενώ εκείνες των χρηστών με τις λειτουργίες που αυτοί θα εκτελούν με τη χρήση του λογισμικού
- β) Οι επιχειρησιακές συνδέονται με τους επιχειρησιακούς στόχους της ομάδας που αναπτύσσει το λογισμικό, ενώ εκείνες των χρηστών με τις λειτουργίες που αυτοί θα εκτελούν με τη χρήση του λογισμικού
- γ) Οι επιχειρησιακές μελετούν το οργανόγραμμα του οργανισμού, για τον οποίο αναπτύσσεται το λογισμικό, ενώ εκείνες των χρηστών τις ανάγκες κατάρτισης αυτών
- δ) Όλα τα παραπάνω

**18. Μια λειτουργική απαίτηση (functional requirement) λογισμικού:**

- α) Περιγράφει μια αλληλεπίδραση του συστήματος και του περιβάλλοντός του
- β) Προσδιορίζει τη γλώσσα προγραμματισμού του κώδικα του λογισμικού
- γ) Προσδιορίζει την ταχύτητα εκτέλεσης μιας εργασίας που υλοποιεί το λογισμικό
- δ) Όλα τα παραπάνω

**19. Ο πρωτεύον actor μιας περίπτωσης χρήσης (use case):**

- α) Την εκκινεί
- β) Είναι ο κυρίως ενδιαφερόμενος για τα αποτελέσματά της
- γ) Είναι κάποιος από τους ενδιαφερόμενους (stakeholders) του συστήματος
- δ) Όλα τα παραπάνω

**20. Η κλάση συσχέτισης (association class) μας επιτρέπει να προσθέσουμε:**

- α) Ιδιότητες και λειτουργίες σε μια συσχέτιση
- β) Μόνο ιδιότητες σε μια συσχέτιση
- γ) Μόνο λειτουργίες σε μια συσχέτιση
- δ) Τίποτα από τα παραπάνω

**21. Το μοντέλο πεδίου (domain model) στην αντικειμενοστρεφή ανάλυση απεικονίζει:**

- α) Τις έννοιες του προβλήματος που εξετάζουμε
- β) Τις κλάσεις του λογισμικού
- γ) Την επικοινωνία των αντικειμένων του λογισμικού
- δ) Τις λειτουργίες των κλάσεων που αντιστοιχούν στις έννοιες του προβλήματος που εξετάζουμε

**22. Αρμοδιότητα (responsibility) μιας κλάσης είναι:**

- α) Η υποχρέωση που έχουν τα αντικείμενά της να εκτελούν μια λειτουργία ή να γνωρίζουν κάποια πληροφορία
- β) Το δικαίωμα που έχει η κλάση να τροποποιεί την τιμή μιας ιδιότητας
- γ) Η περιγραφή των λειτουργιών της
- δ) Η περιγραφή των αντικειμένων που επικοινωνούν με τα στιγμιότυπά της

**23. Οι αφηρημένες κλάσεις (abstract classes):**

- α) Δεν μπορούν να δημιουργήσουν αντικείμενα
- β) Μπορεί να έχουν λειτουργίες αφηρημένες (abstract) ή και συγκεκριμένες (concrete)
- γ) Όλα τα προαναφερθέντα
- δ) Τίποτα από τα παραπάνω

**24. Με τη μεταβίβαση (delegation) στο αντικειμενοστρεφές λογισμικό:**

- α) Μια κλάση μεταβιβάζει δικές της αρμοδιότητες σε κάποια άλλη κλάση, την οποία χρησιμοποιεί μέσω ανταλλαγής μηνυμάτων
- β) Μια κλάση μεταβιβάζει μέρος της κατάστασής (state) της σε υποκλάση της
- γ) Μια κλάση κληρονομεί την υπερκλάση της
- δ) Τίποτα από τα παραπάνω

**25. Ποια είναι η διαφορά της λογικής από τη φυσική αρχιτεκτονική λογισμικού;**

- α) Η λογική ασχολείται με μονάδες λογισμικού (software modules), ενώ η φυσική με εκτελέσιμες οντότητες λογισμικού
- β) Η λογική περιγράφει τις λειτουργίες του λογισμικού, ενώ η φυσική περιγράφει τον τρόπο εκτέλεσης του λογισμικού
- γ) Η λογική επικεντρώνεται στον αλγόριθμο, ενώ η φυσική στον πηγαίο κώδικα
- δ) όλα τα παραπάνω

**26. Το βασικό χαρακτηριστικό των αντικειμενοστρεφών γλωσσών είναι:**

- α) Η ενθυλάκωση (encapsulation)
- β) Ο πολυμορφισμός (polymorphism)
- γ) Η κληρονομικότητα (inheritance)
- δ) Όλα τα παραπάνω

**27. Σύζευξη (coupling) μεταξύ κλάσεων εμφανίζεται όταν:**

- α) Αντικείμενα μιας κλάσης καλούν λειτουργίες της άλλης
- β) Μια κλάση είναι υποκλάση της άλλης
- γ) Κλήσεις αντικειμένων μιας κλάσης έχουν ως παραμέτρους αντικείμενα της άλλης
- δ) Όλα τα παραπάνω

**28. Κατά την αντικειμενοστρεφή σχεδίαση η αρχή της υποκατάστασης (Liskov substitution principle) απαιτεί:**

- α) Κάθε υποκλάση να υποκαθιστά πλήρως την υπερκλάση της
- β) Τα αντικείμενα της υποκλάσης να μπορούν να χρησιμοποιηθούν ως αντικείμενα της υπερκλάσης χωρίς παρενέργειες
- γ) Κάθε υποκλάση να είναι εξειδίκευση της υπερκλάσης
- δ) Όλα τα παραπάνω

**29. Η μετρική της κυκλωματικής πολυπλοκότητας (cyclomatic complexity) μετρά:**

- α) Τον αριθμό των διαφορετικών μονοπατιών στη ροή μιας μονάδας
- β) Τον αριθμό των επαναλήψεων μιας μονάδας
- γ) Τον χρόνο εκτέλεσης μιας μονάδας

δ) Τον αριθμό των επαναλαμβανόμενων σφαλμάτων μιας μονάδας

**30. Η αρχιτεκτονική προσανατολισμένη στις υπηρεσίες (SOA) χαρακτηρίζεται κυρίως από:**

- α) Ισχυρή σύζευξη (Strongly Coupled)
- β) Χαλαρή σύζευξη (Loosely Coupled)
- γ) Υψηλή συνεκτικότητα (Strongly Cohesive)
- δ) Χαμηλή συνεκτικότητα (Loosely Cohesive)

**31. Ποιο από τα παρακάτω αποτελεί βασική αρχή μιας αρχιτεκτονικής;**

- α) Συνέπεια (Consistency)
- β) Αξιοπιστία (Reliability)
- γ) Επεκτασιμότητα (Scalability)
- δ) Όλα τα παραπάνω

**32. Με την SOA (αρχιτεκτονική προσανατολισμένη στις υπηρεσίες) ποια αρχιτεκτονική περιγράφουμε;**

- α) Την αρχιτεκτονική της Εφαρμογής
- β) Την αρχιτεκτονική της Υπηρεσίας
- γ) Την αρχιτεκτονική των Συστατικών (Components)
- δ) Κανένα από τα παραπάνω

**33. \_\_\_\_\_ υπάρχει σαφής διαχωρισμός μεταξύ της παρουσίασης των πληροφοριών και των υπολογιστικών πράξεων που δημιουργούν και επεξεργάζονται αυτές τις πληροφορίες.**

- α) Στις Αρχιτεκτονικές Master-Slave
- β) Στα Συστήματα Client-Server
- γ) Στην Αρχιτεκτονική Client-Server δύο επιπέδων
- δ) Τόσο στις αρχιτεκτονικές Master-Slave όσο και στα συστήματα Client-Server

**34. Ποια αρχιτεκτονική χρησιμοποιείται, όταν υπάρχει μεγάλος όγκος συναλλαγών (transactions) προς επεξεργασία από τον διακομιστή;**

- α) Πολυεπίπεδη (Multi-tier) αρχιτεκτονική πελάτη-διακομιστή (Client-Server)
- β) Αρχιτεκτονική Master-Slave
- γ) Κατανεμημένη αρχιτεκτονική συστατικών (Component)
- δ) Peer-to-Peer αρχιτεκτονική

**35. Ποια αρχιτεκτονική είναι εξαρτημένη από το μεσαίο-λογισμικό (middleware);**

- α) Πολυεπίπεδη (Multi-tier) αρχιτεκτονική πελάτη-διακομιστή (Client-Server)
- β) Αρχιτεκτονική Master-Slave
- γ) Κατανεμημένη αρχιτεκτονική συστατικών (Component)
- δ) Peer-to-Peer αρχιτεκτονική

**36. \_\_\_\_\_ είναι ένας τρόπος παροχής λειτουργικότητας σε έναν απομακρυσμένο διακομιστή με πρόσβαση του πελάτη μέσω ενός προγράμματος περιήγησης ιστού.**

- α) SaaS
- β) SOA
- γ) Διαμορφωσιμότητα (Configurability)
- δ) Τόσο το SaaS όσο και η διαμορφωσιμότητα (Configurability)

**37. Σε ποια αποκεντρωμένη αρχιτεκτονική δεν υπάρχουν διακριτοί πελάτες και διακομιστές;**

- α) Πολυεπίπεδη αρχιτεκτονική πελάτη-διακομιστή (Multi-tier Client–Server Architecture)
- β) Αρχιτεκτονική Master-Slave
- γ) Κατανεμημένη αρχιτεκτονική συστατικών (Distributed Component Architecture)
- δ) Αρχιτεκτονική Peer-to-Peer

**38. Τι χρησιμοποιείται για τη μεταβίβαση αιτημάτων SQL και συσχετισμένων δεδομένων από ένα συστατικό (component) σε ένα άλλο;**

- α) Αλληλεπίδραση SQL πελάτη/διακομιστή (Client/Server SQL Interaction)
- β) Κλήσεις απομακρυσμένης διαδικασίας (RPC)
- γ) Έγχυση SQL (SQL Injection)
- δ) Όλα τα παραπάνω

**39. Όταν ένας πελάτης (client) καλεί μια μέθοδο που περιέχεται σε ένα αντικείμενο που βρίσκεται σε άλλο κόμβο του συστήματος, η CORBA χρησιμοποιεί δυναμική κλήση για να:**

- α) Λάβει σχετικές πληροφορίες από το αποθετήριο διεπαφών σχετικά με την επιθυμητή μέθοδο
- β) Δημιουργήσει μια δομή δεδομένων με παραμέτρους που θα μεταβιβαστούν στο αντικείμενο
- γ) Δημιουργήσει μια αίτηση για το αντικείμενο
- δ) Όλα τα παραπάνω

**40. Ποιες από τις ακόλουθες δηλώσεις είναι εσφαλμένες σε σχέση με τα web-based συστήματα;**

- α) Θα πρέπει να είναι μη κλιμακώσιμα (unscalable)
- β) Πρέπει να είναι σε θέση να αντιμετωπίσουν αβέβαιες, τυχαίες βαριές απαιτήσεις σε υπηρεσίες
- γ) Πρέπει να είναι ασφαλή
- δ) Υπόκεινται σε διάφορους νομικούς, κοινωνικούς και ηθικούς ελέγχους

**41. Ποιος από τους παρακάτω είναι κίνδυνος που σχετίζεται με τη χρήση υπερκειμένου (hypertext) σε εφαρμογές ιστού;**

- α) Απώλεια της αίσθησης της τοποθεσίας και της κατεύθυνσης
- β) Αποπροσανατολισμός της προσοχής των χρηστών από το κεντρικό θέμα
- γ) Όλα τα προαναφερθέντα
- δ) Κανένα από τα παραπάνω

**42. Ποια από τις παρακάτω προτάσεις περιγράφει σαφέστερα τη δοκιμή (testing);**

- α) Εύρεση σπασμένου κώδικα

- β) Αξιολόγηση παραδοτέου για τον εντοπισμό σφαλμάτων
- γ) Ένα στάδιο όλων των έργων
- δ) Κανένα από τα παραπάνω

**43. Ποιο/α από τα ακόλουθα θεωρείται διακριτό επίπεδο δοκιμής;**

- α) Δοκιμή μονάδας
- β) Δοκιμή συστήματος
- γ) Δοκιμές ενσωμάτωσης
- δ) Όλα τα παραπάνω

**44. Η δοκιμή άλφα γίνεται στο:**

- α) Τέλος της ανάπτυξης
- β) Τέλος της σχεδίασης
- γ) Τέλος της κωδικοποίησης
- δ) Κανένα από τα παραπάνω

**45. Το JUnit είναι ένα διαδεδομένο:**

- α) Εργαλείο παραγωγής δεδομένων για δοκιμές
- β) Σύστημα αυτόματης παραγωγής κώδικα
- γ) Framework για την κωδικοποίηση ελέγχων
- δ) Όλα τα παραπάνω

**46. Στην Ανοικτή-κλειστή σχεδίαση (open closed principle) κάθε μονάδα λογισμικού πρέπει να:**

- α) Διαθέτει ένα σημείο επικοινωνίας με το εξωτερικό περιβάλλον
- β) Είναι ανοικτή σε επέκταση και κλειστή σε τροποποίηση
- γ) Βρίσκεται σε διαρκή συντονισμό με τις υπόλοιπες μονάδες
- δ) Αποκρύπτει από τις υπόλοιπες μονάδες τις πληροφορίες που διαχειρίζεται

**47. Ποιο design pattern θα επιλέγατε για να ελαχιστοποιήσετε το κόστος ανάπτυξης επαναχρησιμοποιώντας μεθόδους;**

- α) Adapter Pattern
- β) Singleton Pattern
- γ) Delegation Pattern
- δ) Immutable Pattern

**48. Ποιο design pattern ορίζει εξάρτηση ένα προς πολλά μεταξύ των αντικειμένων;**

- α) Singleton Pattern
- β) Facade Pattern
- γ) Observer Pattern
- δ) Factory Method Pattern

**49. Ποιος είναι ο σκοπός του προτύπου σχεδίασης στρατηγικής (strategy design pattern);**

- α) Η υλοποίηση ενός αλγορίθμου, ο οποίος έχει αμετάβλητα σημεία και ταυτόχρονα σημεία που μεταβάλλονται.



- β) Η δημιουργία αντικειμένων που υλοποιούν κοινή διεπαφή.
- γ) Η χρήση διαφορετικών αλγορίθμων που υλοποιούν διαφορετικές πολιτικές επίλυσης του ίδιου προβλήματος.
- δ) Ο χειρισμός σύνθετων δομών από αντικείμενα με τον ίδιο τρόπο όπως τα μεμονωμένα αντικείμενα.

**50. Ποιο από τα παρακάτω είναι μια γενική δομή που επεκτείνεται για να δημιουργήσει ένα πιο συγκεκριμένο υποσύστημα ή εφαρμογή;**

- α) Επαναχρησιμοποίηση λογισμικού
- β) Αντικειμενοστρεφής γλώσσα προγραμματισμού
- γ) Πλατφόρμα (Framework)
- δ) Κανένα από τα παραπάνω

**51. Ποια από τις παρακάτω γλώσσες προγραμματισμού διαθέτει δυναμικό σύστημα τύπων (dynamically typed language);**

- α) Java
- β) C++
- γ) JavaScript
- δ) C#

**52. Σε ποια από τις ακόλουθες γλώσσες ο κώδικας δεν χρειάζεται μεταγλώττιση προκειμένου να εκτελεστεί;**

- α) C#
- β) Kotlin
- γ) PHP
- δ) Java

**53. Το σημαντικότερο πλεονέκτημα της αρχιτεκτονικής τριών επιπέδων (3 tiers) είναι:**

- α) Η δυνατότητα κλιμακωσιμότητας (scalability)
- β) Η ταχύτητα ανάπτυξης
- γ) Ο διαχωρισμός της διαχείρισης βάσης δεδομένων από τη λογική του συστήματος
- δ) Η ευκολία συντήρησης

**54. Ποια από τις παρακάτω τεχνολογίες υλοποιεί κατανεμημένη αντικειμενοστρεφή αρχιτεκτονική;**

- α) CORBA
- β) DCOM
- γ) .NET Remoting
- δ) Όλες οι παραπάνω

**55. Το πρωτόκολλο SOAP είναι:**

- α) Ένα W3C πρότυπο βασισμένο στη γλώσσα XML
- β) Ένας τρόπος καθαρισμού της μνήμης
- γ) Ένα σύστημα επικοινωνίας υπερυπολογιστών

δ) Ένα πρότυπο σχεδίασης μικροϋπηρεσιών (microservices)

**56. Ως πόρος (resource) μιας υπηρεσίας REST νοείται:**

- α) Οτιδήποτε έχει αρκετό ενδιαφέρον στο πλαίσιο της εφαρμογής που υλοποιεί το λογισμικό
- β) Το σημαντικότερο αντικείμενο του λογισμικού
- γ) Η βασικότερη αφηρημένη έννοια του χώρου του προβλήματος του λογισμικού
- δ) Το σύστημα διευθυνσιοδότησης των υπηρεσιών του λογισμικού

**57. Στην υπηρεσιοστρεφή αρχιτεκτονική (SOA) συνθέτουμε υπηρεσίες:**

- α) Με την ενορχήστρωση (orchestration)
- β) Με τη χορογραφία (choreography)
- γ) Με την ενορχήστρωση ή/και τη χορογραφία
- δ) Με ευριστικές τεχνικές

**58. Στη χορογραφία (choreography):**

- α) Δεν υπάρχει κεντρικός συντονισμός
- β) Η επικοινωνία γίνεται με ανταλλαγή μηνυμάτων
- γ) Οι συμμετέχοντες είναι ομότιμοι (peers)
- δ) Όλα τα παραπάνω

**59. Το μοντέλο IaaS στο υπολογιστικό νέφος μάς προσφέρει:**

- α) Διακομιστές, αποθηκευτικά μέσα, δικτύωση και υπολογιστικούς πόρους
- β) Έτοιμες λύσεις λογισμικού εφαρμογής (application software)
- γ) Εργαλεία ανάπτυξης λογισμικού
- δ) Όλα τα παραπάνω

**60. Το βασισμένο σε μικροϋπηρεσίες (microservices) σχέδιο είναι ένα:**

- α) Σχέδιο που χρησιμοποιείται κυρίως στον συναρτησιακό προγραμματισμό και στον αντικειμενοστρεφή προγραμματισμό
- β) Μικρό πρόγραμμα που αντιπροσωπεύει διακριτή λογική που εκτελείται σε εξειδικευμένο υλικό
- γ) Στυλ σχεδίασης για εταιρικά συστήματα που βασίζεται σε μια χαλαρά συζευγμένη αρχιτεκτονική συστατικών (components) λογισμικού
- δ) Πολύ μικρό κομμάτι κώδικα που δεν υπερβαίνει τις 10 γραμμές

**61. Πότε οι developers ενδείκνυται να χρησιμοποιούν μικροϋπηρεσίες (microservices);**

- α) Όταν θέλουν να γράψουν εφαρμογές κινητού τηλεφώνου που εκτελούνται γρήγορα
- β) Όταν εργάζονται για έργα νανοτεχνολογίας
- γ) Όταν πρέπει να αναπτύξουν μεγάλες εφαρμογές εταιρικού επιπέδου που υπόκεινται σε συχνές αλλαγές
- δ) Όταν αναπτύσσουν εφαρμογές ειδικά για έλεγχο επιστημονικού εξοπλισμού

**62. Πλεονέκτημα της αρχιτεκτονικής της προσανατολισμένης στις μικροϋπηρεσίες (microservices) είναι ότι:**

- α) Οποιαδήποτε μικροϋπηρεσία μπορεί να αλλάξει ανεξάρτητα από τις άλλες

- β) Δεν απαιτείται μεγάλη εμπειρία για να προγραμματιστούν μικροϋπηρεσίες
- γ) Οι μικροϋπηρεσίες είναι τόσο μικρές, που οι προγραμματιστές μπορούν συνήθως να γράψουν πολύ ισχυρές μικροϋπηρεσίες με λίγες γραμμές κώδικα
- δ) Οι μικροϋπηρεσίες είναι εύκολες στη διαχείριση

**63. Ποιες τεχνολογίες χρησιμοποιούνται συνήθως για την υλοποίηση μικροϋπηρεσιών (microservices);**

- α) Docker
- β) Kubernetes
- γ) Όλα τα παραπάνω
- δ) Τίποτα από τα παραπάνω

**64. Πώς εφαρμόζεται μια υπηρεσία υποστήριξης (backing service) στις μικροϋπηρεσίες (microservices);**

- α) Αποτρέπει την αποτυχία μιας μικροϋπηρεσίας
- β) Λειτουργεί ως ειδική υπηρεσία που παρέχει κρίσιμη λειτουργικότητα απαραίτητη για τη μικροϋπηρεσία
- γ) Τερματίζει μια μικροϋπηρεσία, όταν δεν μπορεί πλέον να χειριστεί τον υπολογιστικό φόρτο
- δ) Συντονίζει τη δραστηριότητα δικτύου μεταξύ μικροϋπηρεσιών

**65. Στο υπολογιστικό νέφος (cloud) οι έλεγχοι υγείας (Health checks):**

- α) Σκοπό έχουν την εύρεση των σφαλμάτων στον κώδικα του autoscaler
- β) Σκοπό έχουν την εύρεση των σφαλμάτων στον κώδικα του εξισορροπιστή φορτίου (load balancer)
- γ) Σκοπό έχουν την εύρεση των σφαλμάτων στον κώδικα που διαχειρίζεται τη δημιουργία (κατασκευή) εικονικών μηχανών (VM)
- δ) Είναι ένας μηχανισμός που επιτρέπει στον εξισορροπιστή φορτίου (load balancer) να μάθει εάν ένα στιγμιότυπο υπηρεσίας εκτελείται σωστά



ΑΝΩΤΑΤΟ ΣΥΜΒΟΥΛΙΟ ΕΠΙΛΟΓΗΣ ΠΡΟΣΩΠΙΚΟΥ

ΚΕΝΤΡΙΚΗ ΕΠΙΤΡΟΠΗ ΔΙΑΓΩΝΙΣΜΟΥ

ΠΡΟΚΗΡΥΞΗ 1Γ/2022

ΕΞΕΤΑΖΟΜΕΝΟ  
ΜΑΘΗΜΑ

Σχεδιασμός και Ανάπτυξη Διαδικτυακών Εφαρμογών

ΟΡΘΕΣ ΑΠΑΝΤΗΣΕΙΣ

Ερώτηση	Απάντηση	Ερώτηση	Απάντηση	Ερώτηση	Απάντηση
1	γ	26	δ	51	γ
2	β	27	δ	52	γ
3	α	28	δ	53	α
4	δ	29	α	54	δ
5	β	30	β	55	α
6	α	31	δ	56	α
7	α	32	α	57	γ
8	δ	33	β	58	δ
9	δ	34	α	59	α
10	β	35	γ	60	γ
11	α	36	α	61	γ
12	β	37	δ	62	α
13	α	38	α	63	γ
14	δ	39	δ	64	β
15	γ	40	α	65	δ
16	γ	41	γ		
17	α	42	β		
18	α	43	δ		
19	δ	44	α		
20	α	45	γ		
21	α	46	β		
22	α	47	γ		
23	γ	48	γ		
24	α	49	γ		
25	α	50	γ		

