

# ΑΤΕΙ Πειραιά

## Τμήμα Ηλεκτρονικών Υπολογιστικών Συστημάτων

### Μηχανική Λογισμικού

#### Εξεταστική Εαρινού 2009 – Β' Περίοδος

Διάρκεια εξέτασης: Εβδομήντα πέντε (75) λεπτά

- Μόνο μία (1) απάντηση σε κάθε ερώτηση είναι η σωστή.

- Όλες οι ερωτήσεις είναι ισοδύναμες.

- Όλες οι ερωτήσεις έχουν αρνητική βαθμολογία. Για κάθε εσφαλμένη απάντηση αφαιρείται το μισό ποσοστό του βαθμού που της αναλογεί.

1. Η βασική διαφορά του λογισμικού κατά παραγγελία από το λογισμικό για μαζική χρήση είναι ότι:
  - α. Στο λογισμικό κατά παραγγελία, οι προδιαγραφές ελέγχονται αποκλειστικά από τον κατασκευαστή
  - β. Στο λογισμικό για μαζική χρήση, οι προδιαγραφές ελέγχονται αποκλειστικά από τον πελάτη
  - ▶ γ. Στο λογισμικό για μαζική χρήση, οι προδιαγραφές ελέγχονται αποκλειστικά από τον κατασκευαστή
  - δ. Τίποτε από τα παραπάνω
  
2. Το ίδιο σύνολο προδιαγραφών μπορούν να απασχολήσουν έναν ερασιτέχνη προγραμματιστή για π.χ. τρεις (3) μήνες και έναν επαγγελματία για έξι (6) μήνες. Η βασική διαφορά στις δύο περιπτώσεις σχετίζεται με \_\_\_\_\_ του λογισμικού
  - α. τις προδιαγραφές
  - β. την υποστήριξη
  - ▶ γ. την ποιότητα
  - δ. τη γλώσσα προγραμματισμού
  
3. Τι από τα παρακάτω δεν υπάρχει πάντοτε σε μια διαδικασία ανάπτυξης λογισμικού:
  - α. Εξαγωγή και ανάλυση απαιτήσεων
  - β. Έλεγχος
  - γ. Συντήρηση
  - ▶ δ. Διαχείριση κινδύνων
  
4. Ποιος είναι ο σκοπός της προδιαγραφής μιας δοκιμής (test case specification) στη φάση δοκιμών του λογισμικού;
  - α. Ο προσδιορισμός των βημάτων που πρέπει να γίνουν για την εκτέλεση της δοκιμής
  - ▶ β. Ο προσδιορισμός εισόδων, αναμενόμενων αποτελεσμάτων, και συνθηκών εκτέλεσης για την συγκεκριμένη δοκιμή
  - γ. Ο προσδιορισμός των διαδικασιών που πρέπει να γίνουν για την ολοκλήρωση του λογισμικού με το υλικό
  - δ. Ο προσδιορισμό των δοκιμών που πρέπει να γίνουν για να ολοκληρωθούν μεταξύ τους τα διάφορα τμήματα λογισμικού
  
5. Ένα βασικό ζήτημα για την εφαρμογή διαδικασιών που βασίζονται στο μοντέλο της πρωτοτυποποίησης είναι ότι:
  - α. Τα πρωτότυπα μπορεί να μην αρέσουν στον πελάτη
  - ▶ β. Απαιτούνται εργαλεία για γρήγορη πρωτοτυποποίηση
  - γ. Τα πρωτότυπα μπορεί να μην είναι αρκετά πρωτότυπα

δ. Τα πρωτότυπα πρέπει πάντοτε να περιλαμβάνουν κώδικα

6. Τι από τα παρακάτω αποτελεί αρχή των ευέλικτων (Agile) μεθοδολογιών ανάπτυξης λογισμικού:

- α. Ο έλεγχος του κόστους είναι το παν
- β. Όσο πιο πολλοί προγραμματιστές, τόσο το καλύτερο
- γ. Η λεπτομερής σχεδίαση σε γλιτώνει από πολλά προβλήματα
- δ. Οι αλλαγές είναι ευπρόσδεκτες

7. Ένας πελάτης σας, κατά την διάρκεια αρχικών συζητήσεων για κάποιο έργο λογισμικού, σας ζητά η εφαρμογή που θα αναπτύξετε να «..είναι ασφαλής σύμφωνα με την ισχύουσα νομοθεσία». Πρόκειται για:

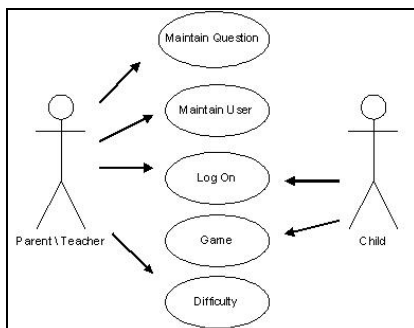
- α. Μη λειτουργική απαίτηση
- β. Λειτουργική απαίτηση
- γ. Και τα δύο, ανάλογα από ποια σκοπιά το βλέπει κανείς
- δ. Τίποτε από τα παραπάνω

8. Η χρήση φυσικής γλώσσας κατά την καταγραφή των απαιτήσεων είναι μη βέλτιστη γιατί:

- α. Οι άνθρωποι έχουν διαφορετικές μητρικές γλώσσες
- β. Ο πολύς διάλογος δεν βοηθάει στην εξαγωγή των απαιτήσεων
- γ. Εισάγεται ασάφεια στην περιγραφή των απαιτήσεων
- δ. Όλα από τα παραπάνω

9. Η δημιουργία σεναρίων ελέγχου αποδοχής για ένα σύστημα (System Acceptance Tests – SATs) μπορεί να βασιστεί στα διαγράμματα:

- α. κατάστασης
- β. αντικειμένων
- γ. περιπτώσεων χρήσης
- δ. ανάπτυξης



10. Το παραπάνω διάγραμμα δεν περιλαμβάνει:

- α. Ρόλους
- β. Όρια συστήματος
- γ. Περιπτώσεις χρήσης
- δ. Συσχετίσεις

11. Η μετάβαση από την σχεδίαση στον πηγαίο κώδικα είναι πιο άμεση με βάση τα διαγράμματα \_\_\_\_\_:

- α. κλάσεων
- β. ανάπτυξης
- γ. περιπτώσεων χρήσης
- δ. πακέτων

12. Τα διαγράμματα ακολουθίας περιλαμβάνουν, μεταξύ άλλων:

- α. ιδιότητες
- β. καταστάσεις και μεταβάσεις
- γ. αντικείμενα και κλήσεις μεθόδων
- δ. πακέτα και κλάσεις

13. Σε ένα καταναμημένο σύστημα τύπου p2p:

- ▶ α. Υπάρχουν συγκεκριμένοι κόμβοι που είναι servers και συγκεκριμένοι που είναι clients
- β. Κάθε κόμβος πότε είναι client και πότε είναι server
- γ. Δεν υφίστανται οι έννοιες client και server
- δ. Τίποτε από τα παραπάνω

14. Τι από τα παρακάτω δεν αποτελεί επιθυμητή ιδιότητα ενός καταναμημένου συστήματος:

- α. Ευκολία στην επέκταση
- β. Ανοχή στις βλάβες
- γ. Κατανομή εργασιών σε πολλές μηχανές
- ▶ δ. Κοινό λειτουργικό σύστημα σε κάθε μηχανή

15. Μια εφαρμογή client-server όπου το client τμήμα είναι γραμμένο σε γλώσσα Java (Java applet) μπορεί να θεωρηθεί:

- α. Εφαρμογή thin client
- β. Εφαρμογή fat client
- ▶ γ. Λίγο και από τα δύο
- δ. Αυτόνομη εφαρμογή

16. Έστω ένα πεδίο σε ένα παράθυρο διαλόγου μέσω του οποίου εισάγονται ακέραιοι από το 0 έως το 20. Κατά την διαδικασία ελέγχου μαύρου κουτιού, η καταλληλότερη ισοδύναμη διαμέριση είναι:

- ▶ α.  $0 < k < 20$  για μια τιμή του  $k$ ,  $k=0$ ,  $k=20$ ,  $k=21$ ,  $k$  αρνητικός
- β.  $0 < k < 20$  για μια τιμή του  $k$
- γ.  $0 < k < 20$  για μια τιμή του  $k$ ,  $k=0$ ,  $k=20$ ,  $k=21$
- δ.  $0 < k < 20$  για μια τιμή του  $k$ ,  $k=0$ ,  $k=20$

17. Έστω ο ακόλουθος κώδικας:

```
1. public static void main(String [] args) {
2.     String s1 = args[0];
3.     String s2 = args[1];
4.     if (s1.equals(s2))
5.         System.out.println("Equal");
6.     else
7.         System.out.println("Not equal");
8. }
```

Εφόσον καλείται πάντοτε με δύο ορίσματα τότε τα δυνατά μονοπάτια εκτέλεσης είναι:

- α. 1,2,3,4,5,6,7,8
- β. 1,2,3,4,5,7,8 και 1,2,3,5,8
- ▶ γ. 1,2,3,4,5,8 και 1,2,3,4,6,7,8
- δ. Κανένα από τα παραπάνω

18. Στον κώδικα του προηγούμενου παραδείγματος το ποσοστό κάλυψης που παίρνουμε αν κάνουμε basic block coverage μόνο με `args[0] == args[1]` είναι:

- ▶ α. 100%
- β. 50%
- γ. 25%
- δ. 0%

19. Ο έλεγχος ολοκλήρωσης (integration testing) αφορά:

- α. Αυτόνομα Συστήματα
- β. Συστήματα Πραγματικού Χρόνου
- γ. Συστήματα ενός χρήστη
- ▶ δ. Συστήματα αποτελούμενα από υποσυστήματα

20. Ο έλεγχος ολοκλήρωσης (integration testing) ανήκει στην κατηγορία ελέγχων:

- ▶ α. Γυάλινου κουτιού
- β. Μαύρου κουτιού
- γ. Διεπαφής χρήστη
- δ. Επιδόσεων

21. Όταν ένα έργο λογισμικού δεν πηγαίνει καλά, τότε τι από τα παρακάτω δεν αποτελεί μια πιθανή λύση:

- α. Αλλαγή του Υπευθύνου Έργου
- β. Αλλαγή μεθοδολογίας
- γ. Συζήτηση με τον πελάτη για παροχή παράτασης
- ▶ δ. Ξαναγράψιμο του κώδικα από την αρχή

22. Η διαστασιολόγηση λογισμικού κατ' αναλογία αφορά λογισμικό:

- α. Που δεν έχουμε αναπτύξει στο παρελθόν
- β. Για το οποίο γνωρίζουμε το ανώτατο όριο χρημάτων που μπορεί να δώσει ο πελάτης
- ▶ γ. Που έχουμε αναπτύξει πολλές φορές στο παρελθόν
- δ. Για το οποίο δεν έχουμε αρκετά στοιχεία

23. Η διαστασιολόγηση λογισμικού με βάση τα Function Points (FPs) εξαρτάται από:

- ▶ α. Τις διάφορες λειτουργίες της εφαρμογής
- β. Την γλώσσα προγραμματισμού
- γ. Την πλατφόρμα λειτουργίας
- δ. Την μεθοδολογία ανάπτυξης λογισμικού

24. Έστω μια εφαρμογή έχει υπολογιστεί ότι θα έχει μέγεθος 33.750 γραμμές κώδικα. Αν εφαρμοστεί η μέθοδος COCOMO τότε υπολογίζεται ότι το έργο απαιτεί προσπάθεια \_\_\_\_\_ ανθρωπομηνών:

(Δίδεται:  $SM = \alpha * (KLOC)^\beta$  όπου  $\alpha=2,4$  και  $\beta=1$ )

- ▶ α. 81000
- β. 81
- γ. 78000
- δ. 90

25. Για την εφαρμογή του προηγούμενου ερωτήματος, η διάρκεια του έργου σε ανθρωπομήνες με βάση την μέθοδο COCOMO είναι:

(Δίδεται:  $TDEV = \alpha * (SM)^\beta$  όπου  $\alpha=2,5$  και  $\beta=0,5$ )

- ▶ α. 711,512
- β. 23,72
- γ. 698,212
- δ. 22,5