

**1. Một đồ thị chương trình có 9 cạnh và 8 nút. Độ phức tạp (cyclomatic complexity) của đồ thị này là bao nhiêu?**

☐ 9

☐ 8

☐ 2

☒ 3

**2. Phương pháp review nào được áp dụng đặc trưng cho lập trình cực đoan:**

☐ Inspection

☐ Walkthrough

☒ Pair programming/review

☐ Technical review

**3. Công cụ kiểm thử đơn vị nào sau đây giúp kiểm tra , phân tích thông lượng tải và xác định các vùng có vấn đề**

☐ Simulator

☐ Sinh tải (traffic generator)

☐ Phân tích bao phủ test

☒ Phân tích mạng (network analyzer)

**4. Framework nào được sử dụng cho kiểm thử đơn vị trên nền C#**

☐ CPPUnit

☐ JUnit

☒ NUnit

☐ Funit

**5. Thành phần nào sau đây được sử dụng nếu bộ phận được test cần gọi hoặc lấy tham số từ một phần chương trình khác.**

☒ Stub

☐ Test driver

☐ Stubbon

☐ Mutant

**6. Độ đo nào sau đây không được thu thập trong quá trình code review**

- ☒ Số người tham gia
- ☐ Số dòng code trên 1h
- ☐ Số yêu cầu thay đổi trên 1000 dòng lệnh
- ☐ Thời gian cần cho quá trình code review

**7. Ghép các phương pháp kiểm thử tĩnh và tính chất đặc trưng của chúng**

1. tác giả dẫn dắt	s) inspection
2. không cần xây dựng tài liệu	t) peer review
3. không có tham gia quản lý	u) informal review
4. dẫn dắt bởi moderator hoặc leader	v) walkthrough
5. sử dụng tiêu chuẩn đầu và cuối	

- ☒ s = 4 and 5, t = 3, u = 2, v = 1
- ☐ s = 4, t = 3, u = 2 and 5, v = 1
- ☐ s = 1 and 5, t = 3, u = 2, v = 4
- ☐ s = 4 and 5, t = 1, u = 2, v = 3

**8. Hàm nào sau đây thích hợp nhất cho kiểm thử với kết quả float**

- ☐ assertEquals(Object expected, Object actual)
- ☐ assertEquals(int expected, int actual)
- ☒ assertEquals(double expected, double actual, double tolerance)
- ☐ assertEquals(Object expected, Object actual)

**9. Khẳng định về kiểm thử tĩnh nào là sai ?**

- ☒ Đánh giá cả về code và về người lập trình
- ☐ Bao gồm các kỹ thuật như: reviews, inspection...
- ☐ Thường thực hiện trước kiểm thử động
- ☐ Không cần kiểm thử đối với thực thi mã nguồn

**10. So sánh nào tốt nhất giữa giữa inspection và walkthrough**

- ☐ Cùng được dẫn dắt bởi tác giả
- ☐ Tác giả thuyết trình trong walkthrough còn inspection thì không

- ☐ Cả 2 cùng được dẫn dắt bởi moderator
- ☒ Inspection được dẫn dắt bởi moderator còn walkthrough được dẫn dắt bởi tác giả

#### 11. Thứ tự thực hiện các bước trong quá trình code review

- ☒ Readiness, Preparation, Examination, Rework, Validation
- ☐ Readiness, Preparation, Rework, Validation, Examination
- ☐ Preparation, Examination, Validation, Readiness, Rework
- ☐ Preparation, Examination, Readiness, Rework, Validation

#### 12. Khác nhau giữa driver và stub

- ☒ Driver được dùng trong chiến lược Bottom-Up, Stub được dùng trong chiến lược Top-Down.
- ☐ Driver được dùng trong chiến lược Top-Down, Stub được dùng trong chiến lược Bottom-Up.
- ☐ Stub gọi tới hàm cần test và Driver được tham chiếu bởi hàm cần test
- ☐ Tất cả đều sai

#### 13. Đây là đặc điểm của Stub

- ☐ Stub được dùng trong chiến lược Bottom-Up
- ☒ Stub được dùng cho chiến lược Top-Down
- ☐ Stub gọi đến hàm cần test
- ☐ Tất cả các ý kiến trên đều đúng

#### 14. Trong các loại sau, loại nào không phải là review

- ☐ Inspection
- ☐ Walkthrough
- ☐ Technical review
- ☒ Equivalence partitioning

#### 15. Trong các phương án sau đây, đâu là đặc điểm của kỹ thuật inspection

- ☐ Là quá trình code review mà trong đó có sự tham gia của tác giả viết chương trình.
- ☐ Là quá trình code review mà trong đó không có sự tham gia của tác giả viết chương trình.
- ☐ Là quá trình code review mà trong đó các đồng nghiệp tiến hành kiểm tra chéo lẫn nhau.
- ☒ A và C

**16. Điểm khác nhau cơ bản giữa inspection và walkthrough là gì?**

- ☐ Inspection được dẫn dắt bởi các tác giả trong khi walkthrough được dẫn dắt bởi một người điều hành.
- ☐ Inspection có một người lãnh đạo được đào tạo trong khi walkthrough không có người lãnh đạo.
- ☒ Inspection không có sự tham gia của tác giả trong khi walkthrough lại có sự tham gia của tác giả.
- ☐ Walkthrough được dẫn dắt bởi tác giả trong khi inspection lại được dẫn dắt bởi một nhà điều hành

**17. Trong các loại kiểm thử sau, loại nào là kiểm thử đơn vị động**

- ☐ Kiểm thử luồng dữ liệu
- ☐ Kiểm thử luồng điều khiển
- ☐ Kiểm thử miền giá trị
- ☒ Tất cả các phương án

**18. Lợi ích mà kiểm thử đơn vị tĩnh mang lại là gì?**

- ☐ Phát hiện được lỗi giao diện
- ☒ Giúp phát hiện các vấn đề về logic và mã
- ☐ Phát hiện được các chức năng thiếu hoặc không đúng.
- ☐ Tất cả đều đúng

**19. Kiểm thử hộp đen**

- ☐ Tập trung vào chức năng của hệ thống
- ☐ Người test không cần phải là chuyên gia
- ☐ Giúp xác định mơ hồ và mâu thuẫn trong chức năng cụ thể
- ☒ Tất cả

**20. Khái niệm test suite**

- ☐ Một tập các test cases.
- ☐ Một tập các test cases được thực hiện theo một trình tự.
- ☐ Các test suite thường liên quan đến miền của hệ thống mà nó thực hiện, độ ưu tiên , hoặc nội dung.
- ☒ B,C

**21. Ai chịu trách nhiệm trực tiếp kiểm thử đơn vị**

- ☐ Người phân tích yêu cầu
- ☐ Người thiết kế hệ thống
- ☒ Người lập trình
- ☐ Người kiểm thử

**22. Có bao nhiêu bước trong khi xem xét lại mã chương trình (code review)**

- ☐ 4
- ☐ 5
- ☒ 6
- ☐ 7

**23. Số liệu nào được tổng hợp sau khi xem xét lại mã chương trình?**

- ☐ Số dòng mã được xem xét lại trong 1 ngày
- ☐ Số CR (change request – yêu cầu thay đổi) được sinh ra trên 100 dòng code
- ☐ Số CR được sinh ra trên 1 giờ
- ☒ Tổng số giờ cần cho quá trình xem xét lại mã chương trình

**24. “Một loại lỗi được gọi là lỗi miền được xác định và sau đó các dữ liệu kiểm thử được lựa chọn để bắt các lỗi đó”, đây là kỹ thuật kiểm thử dữ liệu nào?**

- ☐ Kiểm thử luồng điều khiển (Control flow testing)
- ☐ Kiểm thử luồng dữ liệu (Data flow testing)
- ☒ Kiểm thử miền (Domain testing)
- ☐ Kiểm chức năng (Functional testing)

**25. Trong kiểm thử đơn vị động tại sao môi trường của đơn vị phải được giả lập?**

- ☐ Bởi vì môi trường được giả lập thì sẽ cho kết quả kiểm thử chính xác hơn.
- ☒ Bởi vì mỗi đơn vị được kiểm thử trong sự độc lập
- ☐ Bởi vì để kiểm thử đơn vị có phù hợp với các môi trường khác hay không
- ☐ Cả ba đáp án đều đúng

**26. Trong kiểm thử đơn vị động, có bao nhiêu cách giả lập môi trường của đơn vị được kiểm thử?**

- ☒ 2

☐ 3

☐ 4

☐ 5

Đáp án là 2 vì có hai kiểu giả lập là test driver và stubs

**27. Trong Junit, “nếu giá trị mong muốn và giá trị tham chiếu đến cùng một đối tượng trong bộ nhớ thì xác nhận đúng, nếu không thì xác nhận lỗi”, đây là phương thức nào?**

☐ assertEquals(Object expected, Object actual)

☐ assertEquals(int expected, int actual)

☒ assertEquals(Object expected, Object actual)

☐ assertTrue(Boolean condition)

**28. Trong kiểm thử đơn vị, công cụ nào được sử dụng để kiểm tra chất lượng của phần mềm để đảm bảo rằng nó thỏa mãn tiêu chuẩn mã tối thiểu?**

☒ Code auditor

☐ Bound checker

☐ Documenter

☐ In-circuit emulators

**29. Trong kiểm thử đơn vị, công cụ “static code (path) analyzer” được sử dụng để làm gì?**

☒ Xác định đường đi để kiểm thử dựa trên cấu trúc của mã

☐ Đọc mã nguồn và tự động sinh ra các mô tả

☐ Phân tích lưu lượng truy cập và xác định các vấn đề khu vực

☐ Cả ba đáp án

**30. Backtracking là phương pháp dùng để làm j?**

☐ Tìm lỗi

☐ Sửa lỗi (fix)

☒ Gỡ lỗi (debug)

☐ Xác định nguyên nhân gây ra lỗi

**31. Các kỹ thuật cụ thể để làm kiểm thử tĩnh (static unit testing)**

☐ Inspection

☐ Walkthrough

☒ Cả hai đều đúng

☐ Cả hai đều sai

**32. Trong kiểm thử đơn vị, người ta thường sử dụng khái niệm nào sau đây để mô phỏng việc test**

☐ Test driver

☐ Code Review

☐ Stub

☒ Cả a) và c)

**33. Để kiểm tra một hàm, người lập trình phải viết một ....., cái mà gọi hàm được test và thông qua nó để kiểm thử dữ liệu**

☐ Stub

☐ Proxy

☒ Driver

☐ Cả ba đáp án đều sai

**34. Kiểm thử mà không quan tâm tới code của chương trình, người kiểm thử chỉ chạy và quan sát hành vi của chương trình sau đó quan sát hệ thống là loại kiểm thử nào trong các loại sau:**

☒ Kiểm thử động

☐ Kiểm thử đơn vị

☐ Kiểm thử tĩnh

☐ Kiểm thử chức năng

**35. Người kiểm thử sửa chương trình và chạy bộ test xem chương trình có báo lỗi hay thay đổi hay không là bản chất của loại kiểm thử nào sau đây:**

☒ Kiểm thử đột biến

☐ Kiểm thử stress

☐ Kiểm thử chức năng

☐ Kiểm thử chịu tải

**36. Mutation Testing bản chất là sửa chương trình và chạy bộ test xem nó có báo lỗi hay thay đổi chương trình hay không. Khi một chương trình được sửa đi ta gọi là một**

**mutant. Một mutant mà kết quả đầu ra luôn giống với chương trình gốc thì được gọi là gì?**

- ☐ Killable
- ☒ Equivalent
- ☐ stubborn
- ☐ Cả ba đáp án trên đều sai

**37. Chia khóa thành công của việc kiểm thử tĩnh (đọc lại và kiểm tra code):**

- ☒ Chia để trị
- ☐ Thực hiện lần lượt từ dưới lên
- ☐ Thực hiện lần lượt từ trên xuống
- ☐ Nhảy đến đoạn code cần kiểm tra

**38. Kiểm thử không chạy chương trình chỉ dựa vào việc review code là loại kiểm thử nào trong số những kiểm thử sau:**

- ☐ Kiểm thử động
- ☒ Kiểm thử tĩnh
- ☐ Kiểm thử chức năng
- ☐ Kiểm thử đột biến

**39. Chỉ ra khái niệm tương ứng với việc kiểm thử ở mức độ thấp nhất (các phương thức – method, hàm – function, lớp – class trong mã nguồn) nhằm đảm bảo các thành phần trên hoạt động đúng như yêu cầu. Việc kiểm tra ở mức độ này thường do chính các Lập trình viên (Developer) thực hiện trong quá trình mã hóa (coding, implement).**

- ☐ Kiểm thử chức năng
- ☒ Kiểm thử đơn vị
- ☐ Kiểm thử đột biến
- ☐ d) Kiểm thử chịu tải