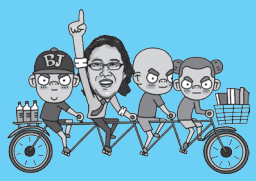
수험생의 마음으로 만든 책! 시나공 시리즈

2020 시나공

시험에 나오는 것만 공부한다!



2020년 2회 정보처리기사 실기 시험 100% 합격전략집

1일차 2일차 3일차 4일차 5일차 6일차 7일차 8일차 9일차 10일차
11일차 **12일차** 13일차 14일차 15일차 16일차 17일차 18일차 19일차 20일차

7장 애플리케이션 테스트 관리

핵심 090 애플리케이션 테스트
핵심 091 애플리케이션 테스트의 분류
핵심 092 화이트박스 테스트 (White Box Test)
핵심 093 블랙박스 테스트 (Black Box Test)
핵심 094 개발 단계에 따른 애플리케이션 테스트
핵심 095 통합 테스트(Integration Test)
핵심 096 애플리케이션 테스트 프로세스
핵심 097 테스트 케이스(Test Case)
핵심 098 테스트 시나리오 (Test Scenario)
핵심 099 테스트 오라클(Test Oracle)
핵심 100 테스트 자동화 도구 유형
핵심 101 결함 관리(Fault)
핵심 102 애플리케이션 성능
핵심 103 소스 코드 최적화

7장 | 애플리케이션 테스트 관리

[핵심090] 애플리케이션 테스트

- 애플리케이션에 잠재되어 있는 결함을 찾아내는 일련의 행위 또는 절차이다.
- 애플리케이션 테스트는 개발된 소프트웨어가 고객의 요구사항을 만족시키는지 확인(Validation)하고 소프트웨어가 기능을 정확히 수행하는지 검증(Verification)한다.
- 애플리케이션 테스트의 기본 원리
  - 완벽한 테스트 불가능 : 애플리케이션 테스트는 소프트웨어의 잠재적인 결함을 줄일 수 있지만 소프트웨어에 결함이 없다고 증명할 수는 없음. 즉 완벽한 소프트웨어 테스팅은 불가능함
  - 결함 집중(Defect Clustering) : 애플리케이션의 결함은 대부분 개발자의 특성이나 애플리케이션의 기능적 특징 때문에 특정 모듈에 집중되어 있으며, 애플리케이션의 20%에 해당하는 코드에서 전체 결함의 80%가 발견된다고 하여 파레토 법칙(Pareto Principle)을 적용하기도 함
  - 살충제 패러독스(Pesticide Paradox) : 애플리케이션 테스트에서는 동일한 테스트 케이스로 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 '살충제 패러독스(Pesticide Paradox) 현상이 발생하며, 살충제 패러독스를 방지하기 위해서 테스트 케이스를 지속적으로 보완 및 개선해야 함
  - 테스팅은 정황(Context) 의존 : 애플리케이션 테스트는 소프트웨어 특징, 테스트 환경, 테스터 역량 등 정황(Context)에 따라 테스트 결과가 달라질 수 있으므로, 정황에 따라 테스트를 다르게 수행해야 함
  - 오류-부재의 궤변(Absence of Errors Fallacy) : 소프트웨어의 결함을 모두 제거해도 사용자의 요구사항을 만족시키지 못하면 해당 소프트웨어는 품질이 높다고 말할 수 없음을 의미함

2020년 1회 기사 실기

1. 애플리케이션 테스트에서 살충제 패러독스(Pesticide Paradox)의 개념을 설명하시오.

답 :

2. 소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙이 무엇인지 쓰시오.

답 :

정답 1. 살충제 패러독스는 동일한 테스트 케이스를 반복하면 더 이상 새로운 결함이 발견되지 않는 현상이다.

2. 파레토 법칙(Pareto Principle)

[핵심091] 애플리케이션 테스트의 분류

- 프로그램 실행 여부에 따른 테스트

정적 테스트	프로그램을 실행하지 않고 명세서나 소스 코드를 대상으로 분석하는 테스트
동적 테스트	프로그램을 실행하여 오류를 찾는 테스트로, 소프트웨어 개발의 모든 단계에서 테스트를 수행할 수 있음

- 테스트 기반(Test Bases)에 따른 테스트

명세 기반 테스트	사용자의 요구사항에 대한 명세를 빠짐없이 테스트 케이스로 만들어 구현하고 있는지 확인하는 테스트
구조 기반 테스트	소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트
경험 기반 테스트	유사 소프트웨어나 기술 등에 대한 테스터의 경험을 기반으로 수행하는 테스트

- 시각에 따른 테스트

검증(Verification) 테스트	개발자의 시각에서 제품의 생산 과정을 테스트하는 것으로, 제품이 명세서대로 완성됐는지를 테스트함
확인(Validation) 테스트	사용자의 시각에서 생산된 제품의 결과를 테스트하는 것으로, 사용자가 요구한대로 제품이 완성됐는지, 제품이 정상적으로 동작하는지를 테스트함

• 목적에 따른 테스트

회복(Recovery) 테스트	시스템에 여러 가지 결함을 주어 실패하도록 한 후 올바르게 복구되는지를 확인하는 테스트
안전(Security) 테스트	시스템에 설치된 시스템 보호 도구가 불법적인 침입으로부터 시스템을 보호할 수 있는지를 확인하는 테스트
강도(Stress) 테스트	시스템에 과도한 정보량이나 빈도 등을 부과하여 과부하 시에도 소프트웨어가 정상적으로 실행되는지를 확인하는 테스트
성능(Performance) 테스트	소프트웨어의 실시간 성능이나 전체적인 효율성을 진단하는 테스트로, 소프트웨어의 응답 시간, 처리량 등을 테스트
구조(Structure) 테스트	소프트웨어 내부의 논리적인 경로, 소스 코드의 복잡도 등을 평가하는 테스트
회귀(Regression) 테스트	소프트웨어의 변경 또는 수정된 코드에 새로운 결함이 없음을 확인하는 테스트
병행(Parallel) 테스트	변경된 소프트웨어와 기존 소프트웨어에 동일한 데이터를 입력하여 결과를 비교하는 테스트

1. 다음은 애플리케이션 테스트의 종류를 설명한 것이다. 서로 관련 있는 것끼리 연결하시오.

① 구조 기반 테스트 •	• ⓐ 프로그램을 실행하지 않고 명세서나 소스 코드를 대상으로 분석하는 테스트
② 동적 테스트 •	• ⓑ 소프트웨어의 실시간 성능이나 전체적인 효율성을 진단하는 테스트
③ 회복 테스트 •	• ⓒ 프로그램을 실행하여 오류를 찾는 테스트
④ 정적 테스트 •	• ⓓ 소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트
⑤ 성능 테스트 •	• ⓔ 시스템에 여러 가지 결함을 주어 실패하도록 한 후 올바르게 복구되는지를 확인하는 테스트

답
①:
②:
③:
④:
⑤:

2. 다음은 애플리케이션 테스트 중 시각에 따른 테스트에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 테스트를 쓰시오.

(①) 테스트는 개발자의 시각에서 제품의 생산 과정을 테스트하는 것으로, 제품이 명세서대로 완성됐는지를 테스트하는 것이고, (②) 테스트는 사용자의 시각에서 생산된 제품의 결과를 테스트하는 것으로, 사용자가 요구한대로 제품이 완성됐는지, 제품이 정상적으로 동작하는지를 테스트한다.

답
①:
②:

정답 1. ①-ⓓ ②-ⓒ ③-ⓔ ④-ⓐ ⑤-ⓑ
2. ① 검증(Verification) ② 확인(Validation)

[핵심092] 화이트박스 테스트(White Box Test)

- 모듈의 원시 코드를 오픈시킨 상태에서 원시 코드의 논리적인 모든 경로를 테스트하여 테스트 케이스를 설계하는 방법이다.
- 원시 코드(Source Code)의 모든 문장을 한 번 이상 실행함으로써 수행된다.
- 모듈 안의 작동을 직접 관찰할 수 있다.
- 화이트박스 테스트의 종류

기초 경로 검사	• 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해주는 테스트 기법 • 테스트 측정 결과는 실행 경로의 기초를 정의하는 데 지침으로 사용됨

제어 구조 검사	• 조건 검사(Condition Testing) : 프로그램 모듈 내에 있는 논리적 조건을 테스트하는 테스트 케이스 설계 기법 • 루프 검사(Loop Testing) : 프로그램의 반복(Loop) 구조에 초점을 맞춰 실시하는 테스트 케이스 설계 기법 • 데이터 흐름 검사(Data Flow Testing) : 프로그램에서 변수의 정의와 변수 사용의 위치에 초점을 맞춰 실시하는 테스트 케이스 설계 기법

- 화이트박스 테스트 검증 기준 : 테스트 케이스들이 테스트에 얼마나 적정한지를 판단하는 기준으로, 문장 검증 기준(Statement Coverage), 분기 검증 기준(Branch Coverage), 조건 검증 기준(Condition Coverage), 분기/조건 기준(Branch/Condition Coverage) 등이 있음

1. 다음이 설명하는 애플리케이션 테스트는 무엇인지 쓰시오.

- 원시 코드의 논리적인 모든 경로를 테스트하는 방법이다.
- Source Code의 모든 문장을 한 번 이상 실행함으로써 수행된다.
- 모듈 안의 작동을 직접 관찰할 수 있다.
- 대표적인 기법에는 기초 경로 검사와 제어 구조 검사가 있다.

답 :

2020년 1회 기능사 실기

2. 다음 〈보기〉에서 내부 소스 코드를 테스트하는 화이트박스 테스트에 속하지 않는 테스트 기법을 모두 골라 쓰시오.

〈보기〉

- 기초 경로 테스트
- 조건 테스트
- 한계값 분석
- 루프 테스트
- 데이터 흐름 테스트
- 비교 테스트

답 :

해설 2. 한계값 분석과 비교 테스트는 블랙박스 테스트에 해당합니다.

정답 1. 화이트박스 테스트(White Box Test)
2. 한계값 분석, 비교 테스트

[핵심093] 블랙박스 테스트(Black Box Test)

- 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고도 한다.
- 사용자의 요구사항 명세를 보면서 테스트하는 것으로, 주로 구현된 기능을 테스트한다.
- 소프트웨어 인터페이스에서 실시되는 테스트이다.
- 블랙박스 테스트의 종류

동치 분할 검사	입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 기법
경계값 분석	입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법으로, 입력 조건의 경계값을 테스트 케이스로 선정하여 검사함
원인 – 효과 그래프 검사	입력 데이터 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석한 다음 효용성이 높은 테스트 케이스를 선정하여 검사하는 기법
오류 예측 검사	과거의 경험이나 확인자의 감각으로 테스트하는 기법
비교 검사	여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트하는 기법

1. 애플리케이션 테스트 중 블랙박스 테스트(Black Box Test)의 개념을 설명하시오.

답 :

2. 다음은 블랙박스 테스트(Black Box Test)의 종류별 특징이다. 괄호(①~③)에 들어갈 알맞은 종류를 쓰시오.

- (①) : 여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트하는 기법이다.
- 원인–효과 그래프 검사 : 입력 데이터 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석한 다음 효용성이 높은 테스트 케이스를 선정하여 검사하는 기법이다.
- (②) : 입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법이다.
- 오류 예측 검사 : 과거의 경험이나 확인자의 감각으로 테스트하는 기법이다.
- (③) : 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 기법이다.

답 :

정답 1. 블랙박스 테스트는 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트이다.

2. ① 비교 검사 ② 경계값 분석 ③ 동치 분할 검사

[핵심094] 개발 단계에 따른 애플리케이션 테스트

- 단위 테스트(Unit Test)
  - 코딩 직후 소프트웨어 설계의 최소 단위인 모듈이나 컴포넌트에 초점을 맞춰 테스트하는 것
  - 구조 기반 테스트 : 프로그램 내부 구조 및 복잡도를 검증하는 화이트박스(White Box) 테스트 시행
  - 명세 기반 테스트 : 목적 및 실행 코드 기반의 블랙박스(Black Box) 테스트 시행
- 통합 테스트(Integration Test) : 단위 테스트가 완료된 모듈들을 결합하여 하나의 시스템으로 완성시키는 과정에서의 테스트를 의미함
- 시스템 테스트(System Test) : 개발된 소프트웨어가 해당 컴퓨터 시스템에서 완벽하게 수행되는가를 점검하는 테스트
- 인수 테스트(Acceptance Test)
  - 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트하는 것
  - 알파 테스트 : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법으로, 테스트는 통제된 환경에서 행해지며, 오류와 사용상의 문제점을 사용자와 개발자가 함께 확인하면서 기록함
  - 베타 테스트 : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법으로, 개발자에 의해 제어되지 않은 상태에서 테스트가 행해지며, 발견된 오류와 사용상의 문제점을 기록하고 개발자에게 주기적으로 보고함

2020년 1, 2회 기사 필기

1. 검증 검사 기법 중 개발자의 장소에서 사용자가 개발자 앞에서 행하는 기법으로, 일반적으로 통제된 환경에서 사용자와 개발자가 함께 확인하면서 수행되는 테스트는 무엇인지 쓰시오.

답 :

2. 애플리케이션 테스트는 소프트웨어의 개발 단계에 따라 통합, 시스템, 인수 테스트 등으로 분류된다. 다음 괄호에 공통적으로 들어갈 알맞은 테스트를 쓰시오.

- ( )는 코딩 직후 소프트웨어 설계의 최소 단위인 모듈이나 컴포넌트에 초점을 맞춰 테스트하는 것으로, 인터페이스, 외부적 I/O, 자료 구조, 독립적 기초 경로, 오류 처리 경로, 경계 조건 등을 검사한다.
- ( )는 구조 기반 테스트와 명세 기반 테스트로 나뉘지만 주로 구조 기반 테스트를 시행한다.

답 :

정답 1. 알파 테스트 2. 단위 테스트(Unit Test)

[핵심095] 통합 테스트(Integration Test)

- 단위 테스트가 끝난 모듈을 통합하는 과정에서 발생하는 오류 및 결함을 찾는 테스트 기법이다.
- 비점진적 통합 방식
  - 단계적으로 통합하는 절차 없이 모든 모듈이 미리 결합되어 있는 프로그램 전체를 테스트하는 방법
  - 규모가 작은 소프트웨어에 유리하며 단시간 내에 테스트가 가능함
  - 오류 발견 및 장애 위치 파악 및 수정이 어려움
  - 빅뱅 통합 테스트 : 모듈 간의 상호 인터페이스를 고려하지 않고 단위 테스트가 끝난 모듈을 한꺼번에 결합시켜 테스트하는 방법
- 점진적 통합 방식
  - 모듈 단위로 단계적으로 통합하면서 테스트하는 방법
  - 오류 수정이 용이하고, 인터페이스와 연관된 오류를 완전히 테스트할 가능성이 높음

- 하향식 통합 테스트(Top Down Integration Test) : 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법
- 상향식 통합 테스트(Bottom Up Integration Test) : 프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법
- 혼합식 통합 테스트 : 하위 수준에서는 상향식 통합, 상위 수준에서는 하향식 통합을 사용하여 최적의 테스트를 지원하는 방식으로, 샌드위치(Sandwich)식 통합 테스트 방법이라고도 함

• 회귀 테스트(Regression Test) : 이미 테스트된 프로그램의 테스팅을 반복하는 것으로, 통합 테스트로 인해 변경된 모듈이나 컴포넌트에 새로운 오류가 있는지 확인하는 테스트

1. 다음 설명에 해당하는 테스트를 쓰시오.

• 단위 테스트가 끝난 모듈을 통합하는 과정에서 발생하는 오류및 결함을 찾는 테스트 기법이다. • 테스트 방법에는 비점진적 통합 방식과 점신적 통합 방식이 있다.

답 :

2. 통합 테스트 중 점진적 통합 방식은 모듈 단위로 단계적으로 통합하면서 테스트하는 방법으로 다음과 같은 종류가 있다. 괄호(①, ②)에 들어갈 알맞은 테스트를 쓰시오.

(①) 통합 테스트	프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법이다.
(②) 통합 테스트	프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
혼합식 통합 테스트	하위 수준에서는 (①) 통합, 상위 수준에서는 (②) 통합을 사용하여 최적의 테스트를 지원하는 방식이다.

답
①:
②:

정답 1. 통합 테스트(Integration Test)
2. ① 상향식(Bottom Up) ② 하향식(Top Down)

[핵심096] 애플리케이션 테스트 프로세스

개발된 소프트웨어가 사용자의 요구대로 만들어졌는지, 결함은 없는지 등을 테스트하는 절차이다.

테스트 계획	프로젝트 계획서, 요구 명세서 등을 기반으로 테스트 목표를 정의하고 테스트 대상 및 범위를 결정함
↓	
테스트 분석 및 디자인	테스트의 목적과 원칙을 검토하고 사용자의 요구 사항을 분석함
↓	
테스트 케이스 및 시나리오 작성	테스트 케이스의 설계 기법에 따라 테스트 케이스를 작성하고 검토 및 확인한 후 테스트 시나리오를 작성함
↓	
테스트 수행	테스트 환경을 구축한 후 테스트를 수행함
↓	
테스트 결과 평가 및 리포팅	테스트 결과를 비교 분석하여 테스트 결과서를 작성함
↓	
결함 추적 및 관리	테스트를 수행한 후 결함이 어디에서 발생했는지, 어떤 종류의 결함인지 등 결함을 추적하고 관리함

1. 애플리케이션 테스트 프로세스는 개발된 소프트웨어가 사용자의 요구대로 만들어졌는지, 결함은 없는지 등을 테스트하는 절차이다. 다음 〈보기〉로 제시된 애플리케이션 테스트 프로세스 절차를 진행 순서대로 기호(㉠~㉥)로 나열하시오.

〈보기〉

㉠ 테스트 분석 및 디자인 ㉡ 테스트 결과 평가 및 리포팅 ㉢ 테스트 수행 ㉣ 테스트 계획 ㉤ 테스트 케이스 및 시나리오 작성 ㉥ 결함 추적 및 관리

답 :

정답 1. ㉣, ㉠, ㉤, ㉢, ㉡, ㉥

[핵심097] 테스트 케이스(Test Case)

- 구현된 소프트웨어가 사용자의 요구사항을 정확하게 준수했는지를 확인하기 위해 설계된 입력 값, 실행 조건, 기대 결과 등으로 구성된 테스트 항목에 대한 명세서로, 명세 기반 테스트의 설계 산출물에 해당된다.
- 테스트 케이스 작성 순서

테스트 계획 검토 및 자료 확보
↓
위험 평가 및 우선순위 결정
↓
테스트 요구사항 정의
↓
테스트 구조 설계 및 테스트 방법 결정
↓
테스트 케이스 정의
↓
테스트 케이스 타당성 확인 및 유지 보수

1. 구현된 소프트웨어가 사용자의 요구사항을 정확하게 준수했는지를 확인하기 위해 설계된 입력 값, 실행 조건, 기대 결과 등으로 구성된 테스트 항목에 대한 명세서로, 명세 기반 테스트의 설계 산출물에 해당되는 것은 무엇인지 쓰시오.

답 :

2. 테스트 케이스는 테스트 전략이나 테스트 계획서 등을 기반으로 하여 다음과 같은 순서로 작성된다. 괄호(①, ②)에 들어갈 알맞은 작업 내용을 쓰시오.

테스트 계획 검토 및 자료 확보 → (①) → 테스트 요구사항 정의 → 테스트 구조 설계 및 테스트 방법 결정 → (②) → 테스트 케이스 타당성 확인 및 유지 보수

답
① :
② :

정답 1. 테스트 케이스(Test Case)
2. ① 위험 평가 및 우선순위 결정 ② 테스트 케이스 정의

[핵심098] 테스트 시나리오(Test Scenario)

- 테스트 케이스를 적용하는 순서에 따라 여러 개의 테스트 케이스들을 묶은 집합으로, 테스트 케이스들을 적용하는 구체적인 절차를 명세한 문서이다.
- 테스트 시나리오에는 테스트 순서에 대한 구체적인 절차, 사전 조건, 입력 데이터 등이 설정되어 있다.
- 테스트 시나리오는 시스템별, 모듈별, 항목별 등과 같이 여러 개의 시나리오로 분리하여 작성해야 한다.
- 각각의 테스트 항목은 식별자 번호, 순서 번호, 테스트 데이터, 테스트 케이스, 예상 결과, 확인 등을 포함해서 작성해야 한다.
- 테스트 시나리오는 유스케이스(Use Case) 간 업무 흐름이 정상적인지를 테스트할 수 있도록 작성해야 한다.

1. 테스트 케이스를 적용하는 순서에 따라 여러 개의 테스트 케이스들을 묶은 집합으로, 테스트 케이스들을 적용하는 구체적인 절차를 명세한 문서는 무엇인지 쓰시오.

답 :

정답 1. 테스트 시나리오(Test Scenario)

[핵심099] 테스트 오라클(Test Oracle)

- 테스트 결과가 올바른지 판단하기 위해 사전에 정의된 참 값을 대입하여 비교하는 기법 및 활동이다.
- **참(True) 오라클** : 모든 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하는 오라클로, 발생된 모든 오류를 검출할 수 있음
- **샘플링(Sampling) 오라클** : 특정한 몇몇 테스트 케이스의 입력 값들에 대해서만 기대하는 결과를 제공하는 오라클
- **추정(Heuristic) 오라클** : 샘플링 오라클을 개선한 오라클로, 특정 테스트 케이스의 입력값에 대해 기대하는 결과를 제공하고, 나머지 입력 값들에 대해서는 추정으로 처리하는 오라클
- **일관성 검사(Consistent) 오라클** : 애플리케이션의 변경이 있을 때, 테스트 케이스의 수행 전과 후의 결과 값이 동일한지를 확인하는 오라클

1. 다음은 테스트 오라클(Test Oracle)의 종류에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 테스트 오라클을 쓰시오.

일관성 검사 오라클	애플리케이션의 변경이 있을 때, 테스트 케이스의 수행 전과 후의 결과 값이 동일한지를 확인하는 오라클이다.
참 오라클	모든 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하는 오라클로, 발생된 모든 오류를 검출할 수 있다.
(①) 오라클	샘플링 오라클을 개선한 오라클로, 특정 테스트 케이스의 입력값에 대해 기대하는 결과를 제공하고, 나머지 입력 값들에 대해서는 추정으로 처리하는 오라클이다.
(②) 오라클	특정한 몇몇 테스트 케이스의 입력 값들에 대해서만 기대하는 결과를 제공하는 오라클이다.

답
①:
②:

정답 1. ① 추정(Heuristic) ② 샘플링(Sampling)

[핵심100] 테스트 자동화 도구 유형

- 사람이 반복적으로 수행하던 테스트 절차를 테스트 자동화 도구를 사용함으로써 휴먼 에러(Human Error)를 줄이고 테스트의 정확성을 유지하면서 테스트의 품질을 향상시킬 수 있다.

※ 휴먼 에러(Human Error) : 사람의 판단 실수나 조작 실수 등으로 인해 발생하는 에러

- 정적 분석 도구(Static Analysis Tools) : 프로그램을 실행하지 않고 분석하는 도구로, 소스 코드에 대한 코딩 표준, 코딩 스타일, 코드 복잡도 및 남은 결함 등을 발견하기 위해 사용됨
- 테스트 실행 도구(Test Execution Tools) : 스크립트 언어를 사용하여 테스트를 실행하는 방법으로, 테스트 데이터와 테스트 수행 방법 등이 포함된 스크립트를 작성한 후 실행함
- 성능 테스트 도구(Performance Test Tools) : 애플리케이션의 처리량, 응답 시간, 경과 시간, 자원 사용률 등을 인위적으로 적용한 가상의 사용자를 만들어 테스트를 수행함으로써 성능의 목표 달성 여부를 확인함
- 테스트 통제 도구(Test Control Tools) : 테스트 계획 및 관리, 테스트 수행, 결함 관리 등을 수행하는 도구로, 종류에는 형상 관리 도구, 결함 추적/관리 도구 등이 있음
- 테스트 하네스 도구(Test Harness Tools) : 테스트가 실행될 환경을 시뮬레이션 하여 컴포넌트 및 모듈이 정상적으로 테스트 되도록 하는 도구

2020년 1회 기능사 실기

1. 인간의 실수 등을 통해 원래의 의도와 다르게 소프트웨어가 예정된 설계를 벗어나 발생하는 오류를 의미하는 용어를 쓰시오.

답 :

2. 프로그램을 실행하지 않고 분석하는 도구로, 소스 코드에 대한 코딩 표준, 코딩 스타일, 코드 복잡도 및 남은 결함 등을 발견하기 위해 사용되는 테스트 자동화 도구는 무엇인지 쓰시오.

답 :

정답 1. 휴먼 에러(Human Error)
2. 정적 분석 도구(Static Analysis Tools)

[핵심101] 결함 관리(Fault)

- 결함은 오류 발생, 작동 실패 등과 같이 소프트웨어가 개발자가 설계한 것과 다르게 동작하거나 다른 결과가 발생되는 것을 의미한다.
- 결함 관리 프로세스 처리 순서 : 결함 관리 계획 → 결함 기록 → 결함 검토 → 결함 수정 → 결함 재확인 → 결함 상태 추적 및 모니터링 활동 → 최종 결함 분석 및 보고서 작성
- 결함 관리 측정 지표

결함 분포	모듈 또는 컴포넌트의 특정 속성에 해당하는 결함 수 측정
결함 추세	테스트 진행 시간에 따른 결함 수의 추이 분석
결함 에이징	특정 결함 상태로 지속되는 시간 측정

• 결함 추적 순서 : 결함 등록(Open) → 결함 검토(Reviewed) → 결함 할당(Assigned) → 결함 수정(Resolved) → 결함 종료(Closed) → 결함 해제(Clarified)

※ Fixed(고정) : 개발자가 필요한 변경 작업을 수행하여 결함 수정 작업을 완료한 상태

• 결함 분류

시스템 결함	애플리케이션 환경이나 데이터베이스 처리에서 발생된 결함
기능 결함	애플리케이션의 기획, 설계, 업무 시나리오 등의 단계에서 유입된 결함
GUI 결함	사용자 화면 설계에서 발생된 결함
문서 결함	기획자, 사용자, 개발자 간의 의사소통 및 기록이 원활하지 않아 발생된 결함

• 결함 심각도

High	더 이상 프로세스를 진행할 수 없도록 만드는 결함
Medium	시스템 흐름에 영향을 미치는 결함
Low	시스템 흐름에는 영향을 미치지 않는 결함

• 결함 우선순위 : 결정적(Critical), 높음(High), 보통(Medium), 낮음(Low) 또는 즉시 해결, 주의 요망, 대기, 개선 권고 등

1. 다음은 테스트에서 발견되는 결함을 유형별로 분류한 것이다. 괄호(①, ②)에 들어갈 알맞은 결함을 쓰시오.

문서 결함	사용자의 온라인/오프라인 매뉴얼의 불일치 등 기획자, 사용자, 개발자 간의 의사소통 및 기록이 원활하지 않아 발생된 결함이다.
(①) 결함	UI 비일관성, 데이터 타입의 표시 오류, 부정확한 커서/메시지 오류 등 사용자 화면 설계에서 발생된 결함이다.
기능 결함	사용자의 요구사항 미반영/불일치, 스크립트 오류, 타시스템 연동 시 오류 등 애플리케이션의 기획, 설계, 업무 시나리오 등의 단계에서 유입된 결함이다.
(②) 결함	시스템 다운, 애플리케이션의 작동 정지, 종료, 응답 시간 지연, 데이터베이스 에러 등 주로 애플리케이션 환경이나 데이터베이스 처리에서 발생된 결함이다.

답
①:
②:

2. 결함 추적은 결함이 발견된 때부터 결함이 해결될 때까지 전 과정을 추적하는 것이다. 다음에 제시된 결함 추적 단계들을 순서대로 기호(㉠~㉥)로 나열하시오..

㉠ 결함 수정(Resolved)	㉡ 결함 등록(Open)
㉢ 결함 해제(Clarified)	㉣ 결함 검토(Reviewed)
㉤ 결함 할당(Assigned)	㉥ 결함 종료(Closed)

답 :

2020년 2회 기능사 실기

3. 결함 관리 과정에서 사용되는 용어인 Fixed의 의미를 쓰시오.

답 :

정답 1. ① GUI ② 시스템 2. ㉡, ㉣, ㉤, ㉠, ㉥, ㉢

3. Fixed는 개발자가 필요한 변경 작업을 수행하여 결함 수정 작업을 완료한 상태이다.

[핵심102] 애플리케이션 성능

• 사용자가 요구한 기능을 최소한의 자원을 사용하여 최대한 많은 기능을 신속하게 처리하는 정도를 나타낸다.

• 애플리케이션 성능 측정 지표

처리량 (Throughput)	일정 시간 내에 애플리케이션이 처리하는 일의 양
응답 시간 (Response Time)	애플리케이션에 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간
경과 시간 (Turn Around Time)	애플리케이션에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간
자원 사용률 (Resource Usage)	애플리케이션이 의뢰한 작업을 처리하는 동안의 CPU 사용량, 메모리 사용량, 네트워크 사용량 등 자원 사용률

2020년 1회 기사 실기

1. 애플리케이션 성능이란 사용자가 요구한 기능을 최소한의 자원을 사용하여 최대한 많은 기능을 신속하게 처리하는 정도를 나타낸다. 애플리케이션 성능 측정의 지표에 대한 설명에서 괄호(①~③)에 들어갈 알맞은 지표를 쓰시오.

자원 사용률	애플리케이션이 의뢰한 작업을 처리하는 동안의 CPU 사용량, 메모리 사용량, 네트워크 사용량 등 자원 사용률
(①)	애플리케이션에 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간
(②)	일정 시간 내에 애플리케이션이 처리하는 일의 양
(③)	애플리케이션에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간

답
①:
②:
③:

정답 ① 응답 시간(Response Time) ② 처리량(Throughput) ③ 경과 시간(Turn Around Time)

[핵심 103] 소스 코드 최적화

- 나쁜 코드(Bad Code)를 배제하고, 클린 코드(Clean Code)로 작성하는 것이다.
- **클린 코드(Clean Code)** : 누구나 쉽게 이해하고 수정 및 추가할 수 있는 단순, 명료한 코드
- **나쁜 코드(Bad Code)** : 코드의 로직이 서로 얽혀 있는 스파게티 코드 등 프로그램의 로직(Logic)이 복잡하고 이해하기 어려운 코드
- 나쁜 코드로 작성된 애플리케이션의 코드를 클린 코드로 수정하면 애플리케이션의 성능이 개선된다.
- **클린 코드 작성 원칙** : 가독성, 단순성, 의존성 배제, 중복성 최소화, 추상화
- **소스 코드 최적화 유형**
  - 클래스 분할 배치 : 하나의 클래스는 하나의 역할만 수행하도록 응집도를 높이고, 크기를 작게 작성함
  - Loosely Coupled(느슨한 결합) : 인터페이스 클래스를 이용하여 추상화된 자료 구조와 메소드를 구현함으로써 클래스 간의 의존성을 최소화함
  - 코딩 형식 준수, 좋은 이름 사용, 적절한 주석문 사용

1. 소스 코드 최적화와 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

- 소스 코드 최적화는 (①)를 배제하고, (②)로 작성하는 것이다.
- (②)는 누구나 쉽게 이해하고 수정 및 추가할 수 있는 단순, 명료한 코드를, (①)는 프로그램의 Logic이 복잡하고 이해하기 어려운 코드를 의미한다.

답
①:
②:

2. 소스 코드 최적화 유형 중 인터페이스 클래스를 이용하여 추상화된 자료 구조와 메소드를 구현함으로써 클래스 간의 의존성을 최소화 하는 것을 의미하는 용어를 쓰시오.

답 :

정답 1. ① 나쁜 코드(Bad Code) ② 클린 코드(Clean Code)
2. Loosely Coupled(느슨한 결합)