



디버깅 및 테스트

Python 기본 사항

발표자 이름

날짜



디버깅

학습 내용



강의 핵심 내용

학습 내용:

- 디버깅과 코드 테스트의 목적을 설명합니다.
- 개발자가 디버거를 사용하여 코드에서 버그를 찾는 방법을 설명합니다.
- Python 코드의 정적 분석을 수행하는 방법을 인식합니다.
- 동적 분석을 구현하는 방법을 찾기 위해 Python 코드를 평가하는 방법을 인식합니다.

디버깅이란?

로깅은 코드의 논리적 문제를 찾는 데 유용한 도구입니다.

디버깅은 코드 자체의 결함을 발견하기 위해 수행합니다. 개념이 간단하며 다음의 두 부분으로 구성됩니다.

디버깅은 실제로 수행할 때 복잡할 수 있습니다. 모든 언어에는 이 문제를 경감하기 위한 디버깅 도구가 있습니다.

코드에서 오류 발견

발견한 오류 수정

PDB: Python 디버거

Python 디버거는 다음 코드를 입력하면 활성화됩니다.

```
python -m pdb <filename>
```

코드의 첫 줄이 실행되면 프롬프트(pdb)가 나타납니다.

이제 몇 가지 명령을 사용할 수 있습니다.

이 도구는 모두 오류를 발견하고 개발자가 오류의 원인을 찾는 데 사용됩니다.

디버깅의 유형

정적 분석

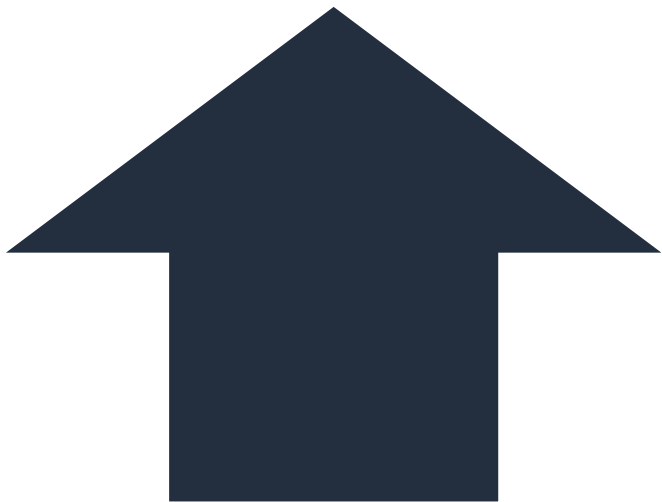
동적 분석

정적 분석

정적 분석의 특징:

- 개발 프로세스에서 지속적으로 수행할 수 있습니다.
- Python 인터프리터는 구문과 의미를 보고하기 때문에 정적 분석 레벨을 포함합니다. 통합 개발 환경(IDE)은 코드를 작성하는 동안 문제를 발견하는 데 도움이 됩니다.
- 중첩과 함수 호출, 코드 복잡성 등의 요인을 고려할 수 있습니다.

정적 분석: 장단점



정적 분석의 장점:

- 코드 문제의 정확한 위치를 파악할 수 있습니다.
- 수정을 처리하는 시간이 더 짧습니다.
- 이후 테스트에서 문제가 더 적게 발생합니다.
- 버그를 초기에 발견하면 비용을 절감할 수 있습니다.



정적 분석의 단점:

- 수동 분석으로 인해 시간이 많이 걸립니다.
- 자동화 도구에서 거짓 양성과 거짓 음성이 나올 수 있습니다.
- 자동화로 인해 보안에 소홀해질 수 있습니다.
- 자동화는 도구를 설정하는 데 사용되는 파라미터와 마찬가지로.

데모: 정적 디버깅



지침

1. **debug.py**라는 파일을 만듭니다.
2. 배운 주제를 토대로, 이 파일에서 기본 Python 프로그램을 작성합니다. 이때 프로그래밍 오류를 포함해야 합니다. 구문 오류나 철자 오류여도 되고, 그 외 다른 오류여도 됩니다.
3. 프로그램을 실행하고 오류가 어떻게 표시되는지 보여 줍니다.

동적 분석

동적 분석의 특징:

- 실행 중인 애플리케이션을 분석함으로써 수행합니다.
- 대부분의 Python용 IDE에는 '디버깅' 모드가 있습니다. 이 모드에서 애플리케이션이 실행되면 개발자는 특정 코드 줄이 호출되거나, 변수에 값(같음, 미만, 초과)이 포함될 때까지 코드를 '한 번에 한 줄씩' 실행합니다.
- 동적 분석 중에 중요한 기법은 로그 파일의 실행 중인 애플리케이션에서 나타나는 값과 조건을 모두 작성하는 것입니다.

동적 분석: 장단점



동적 분석의 장점:

- 런타임 환경(예: 테스트 서버 또는 소프트웨어의 라이브 버전)에서 문제를 발견합니다.
- 실제 코드를 관찰하지 못하는 상황에서도 애플리케이션을 분석할 수 있습니다.
- 정적 분석에서 발견한 거짓 음성이 실제로 음성인지, 아니면 거짓 음성인지 판단할 수 있습니다.
- 모든 애플리케이션에 사용할 수 있습니다.



동적 분석의 단점:

- 자동화로 인해 보안에 소홀해질 수 있습니다.
- 전체 범위를 포함하도록 보장할 수 없습니다.
- 코드 문제를 격리하기가 더 어려울 수 있습니다.

어설션

- 어설션은 if 스테이트먼트와 같은 조건으로 애플리케이션에 있는 값을 확인합니다.
- 동적 분석에서는 런타임 중에 어설션 스테이트먼트를 사용하여 특정 조건이 충족되었을 때 오류를 보고합니다.
- 아래의 함수를 예로 들어 보겠습니다. 개발자가 연령 값이 항상 0보다 큰 양수인지 확인하려고 하는 경우, 다음 어설션로 수를 확인합니다.

```
def loguserage(age):  
    assert age <= 0, "Invalid age was supplied"
```

로그 모니터링

개발자는 로그 파일이라는 텍스트 파일에 코드를 작성합니다. 실행 중인 애플리케이션에서 특정 조건이 충족되면 로깅 코드가 로그 파일에 정보를 작성합니다.

로그 모니터링의 역할은 무엇입니까?

실행 중인 프로그램에서 오류 추적

나중에 검토하기 위해 프로그램이
마지막으로 실행된 시기를 기록해 두면
특정 영역에서 어떤 점이 잘못되었는지
확인할 수 있음

로그 모니터링을 사용하면 실행 중인 애플리케이션을 전체적으로 파악할 수 있습니다. 사용자가 애플리케이션을 사용 중일 때에도 개발자는 애플리케이션이 '실제로' 어떻게 사용되는지 로그 파일을 검사할 수 있습니다.

로그할 내용

다음과 같이 애플리케이션에서 중요한 이벤트를 모두 고려합니다.

- 어디서 발생했는가?
- 언제 발생했는가?
- 인수는 무엇인가?
- 중요한 리소스의 상태는 어떠한가?

오류가 발생할 때 모든 정보를 포착합니다.

- 모든 인수
- 예외와 내부 예외
- 객체 트레이스백: 스택 트레이스

로그 모니터링 도구

Python에서는 기본적으로 logging 라이브러리가 네이티브 로그 모니터링 도구입니다.

라이브러리에 액세스하는 방법

Python 파일 상단에 `import logging`을 입력합니다.

- 이 명령을 사용하면 사용자 지정 가능한 오류 메시지에 액세스하여 서로 다른 우선순위 레벨을 할당한 후 나중에 검토하기 위해 파일에 저장할 수 있습니다.

출력을 파일에 저장하는 방법

Logging 라이브러리를 사용하기 전 아래의 코드 줄을 추가합니다.

```
logging.basicConfig(filename="app.log",  
level=logging.DEBUG)
```

데모: 동적 디버깅



지침

1. **debug.py** 파일을 엽니다.
2. 이 파일을 수정하여 사용자가 입력할 수 있도록 합니다.
3. 사용자가 입력하는 값이 0보다 큰지 여부를 동적으로 테스트할 수 있는 방법은 무엇입니까?



소프트웨어 테스트

학습 내용

강의 핵심 내용

학습 내용:

- 소프트웨어 테스트가 필요한 이유를 설명합니다.



테스트



단위 테스트

단위는 소프트웨어에서 테스트할 수 있는 가장 작은 부분으로, 테스트의 가장 기본 레벨입니다. 단위에는 보통 입력이 한 개 이상, 출력이 한 개 있습니다.

단위 테스트의 예로는 프로그램의 개별 함수를 각각 검증하는 경우를 들 수 있습니다. 개발자는 각자 단위 테스트를 수행할 책임이 있습니다.

통합 테스트

개별 단위를 결합하여 그룹으로 테스트할 수 있습니다. 문제를 발견할 수 있도록 소프트웨어의 서로 다른 부분이 어떻게 상호 작용하는지 테스트하는 것입니다.

비유: 펜을 만들 때 뚜껑, 펜대, 잉크 카트리지 등 펜의 모든 부품(단위)을 개별적으로 만들어 각각 테스트합니다(단위 테스트). 하나 이상의 단위가 준비되면 이들이 서로 어떻게 상호 작용하는지 확인하기 위해 테스트합니다.

개발자가 통합 테스트를 수행할 수도 있지만, 이 테스트는 대개 전담 테스터가 진행합니다.

시스템 테스트

완전히 통합된 애플리케이션을 테스트합니다. 이 레벨에서는 소프트웨어가 특정 요구 사항을 충족하는지 판단합니다.

비유(계속): 펜을 조립하고 나면 펜이 잘 작동하는지 테스트합니다. 올바른 색상으로 작성되는지?, 사양에 맞는 크기인지? 확인합니다.

승인 테스트

승인 테스트는 사용자 요구 사항과 비즈니스 요구 사항, 그리고 소프트웨어를 최종 사용자에게 제공해도 되는지 여부를 고려하는 공식 테스트입니다.

학습 내용 확인

디버깅의 목적은 무엇입니까?

PDB를 사용하는 방법을 배워야 하는 이유는 무엇입니까?

작은 프로그램에는 정적 테스트와 동적 테스트 중 어떤 유형이 더 적합합니까?

단위 테스트와 통합 테스트의 차이점은 무엇입니까?

요점



- 코드의 지속적 통합에는 스타일 확인 작업과 다양한 테스트가 포함됩니다.
- 로그를 사용하여 런타임 시 프로그램에 오류가 발생하는 경우를 포착합니다.
- 디버깅은 프로그램의 오류를 찾는 데 사용되며, 정적 방식과 동적 방식이 있습니다.
- 다음과 같이 다양한 유형의 테스트가 애플리케이션이 예상대로 동작하는지 확인하기 위해 수행됩니다.
 - 단위 테스트
 - 통합 테스트
 - 시스템 테스트
 - 승인 테스트