

Zadanie: Testowanie Jednostkowe z PyTest dla Projektu MediScan

Jarek Hryszko

Scenariusz

Jesteś nowym wolontariuszem w projekcie open source **MediScan** – prostym, ale ważnym narzędziu służącym do analizy podstawowych wyników badań krwi dla szpitali w krajach rozwijających się. Ten projekt jest używany przez personel medyczny z ograniczonym dostępem do zaawansowanego sprzętu diagnostycznego.

Program MediScan został stworzony przez międzynarodową organizację lekarzy i programistów. Obecnie jest używany w 15 krajach, gdzie pomaga w szybkiej interpretacji wyników badań krwi i identyfikacji potencjalnych problemów zdrowotnych.

Opis projektu MediScan

MediScan to prosta biblioteka Pythona, która:

- Analizuje podstawowe parametry badań krwi
- Porównuje je z wartościami referencyjnymi
- Generuje prosty raport wskazujący odchylenia od normy
- Sugeruje możliwe interpretacje kliniczne

Twoje zadanie

Zespół MediScan pracuje nad nową funkcjonalnością obliczania bardziej zaawansowanych wskaźników z podstawowych parametrów krwi. Niedawno dodano moduł `bloodwork_calculator.py`, który zawiera funkcje do obliczania wskaźników takich jak BMI pacjenta czy stosunek neutrofili do limfocytów (NLR).

Problem: Dział kontroli jakości zgłosił, że w module `bloodwork_calculator.py` występują defekty – niektóre obliczenia dają nieprawidłowe wyniki w określonych przypadkach. Zespół potrzebuje testów jednostkowych, aby zidentyfikować i naprawić problemy.

Twoje zadanie obejmuje:

1. Sklonowanie repozytorium projektu MediScan z GitHub
2. Zapoznanie się z kodem modułu `bloodwork_calculator.py`
3. Napisanie testów jednostkowych przy użyciu PyTest, które:
 - Sprawdzą poprawność działania istniejących funkcji
 - Zidentyfikują defekty w obliczeniach
 - Będą wykorzystywać parametryzację testów do sprawdzenia różnych scenariuszy
 - Będą używać fixtures do przygotowania przykładowych danych pacjentów
4. Zaproponowanie poprawek do znalezionych defektów

Instrukcje

1. Sklonuj repozytorium projektu z adresu: `https://github.com/[username]/mediscan-project`
2. Zapoznaj się ze strukturą projektu, szczególnie z modułem `bloodwork_calculator.py`
3. Utwórz plik `test_bloodwork_calculator.py` w katalogu `tests/`
4. Napisz testy jednostkowe zgodnie z wymaganiami
5. Uruchom testy za pomocą PyTest i zidentyfikuj defekty
6. Zaproponuj poprawki do defektów w kodzie

Kryteria oceny

- Poprawność i kompletność testów
- Wykorzystanie różnych funkcji PyTest (parametryzacja, fixtures)
- Jasne komunikaty o błędach w testach
- Prawidłowa identyfikacja defektów w kodzie
- Jakość zaproponowanych poprawek

Wskazówki

- W module `bloodwork_calculator.py` znajdują się co najmniej 3 defekty
- Zwróć uwagę na przypadki brzegowe (np. wartości zerowe, ujemne, ekstremalne)
- Pamiętaj o konwencjach nazewnictwa testów w PyTest
- Używaj markerów PyTest do kategoryzacji testów

Powodzenia!