Họ và tên: Chu Quang Tú

Mã sinh viên: 21020393

**Đề bài:** Nhập vào 2 số dương là chiều dài và chiều rộng của hình chữ nhật. Tính diện tích hình chữ nhật đó

Link github: https://github.com/ChuTu15/kiemthu.git

# **Xây dựng đồ thị điều khiển cho chương trình.**

A screenshot of a cell phone

Description automatically generatedĐơn vị chương trình là một hàm có tên là area nhận vào hai tham số là chiều dài và chiều rộng của hình chữ nhật và trả về diện tích của nó. Đồ thị điều khiển có thể được vẽ như sau:

# **Tìm đường đi trong đồ thị điều khiển thoả mãn**

Với độ phủ C2, ta cần tìm ít nhất hai đường đi, một đường đi cho trường hợp điều kiện “length and width are positive” là đúng và một đường đi cho trường hợp điều kiện là sai. Hai đường đi được chọn là:

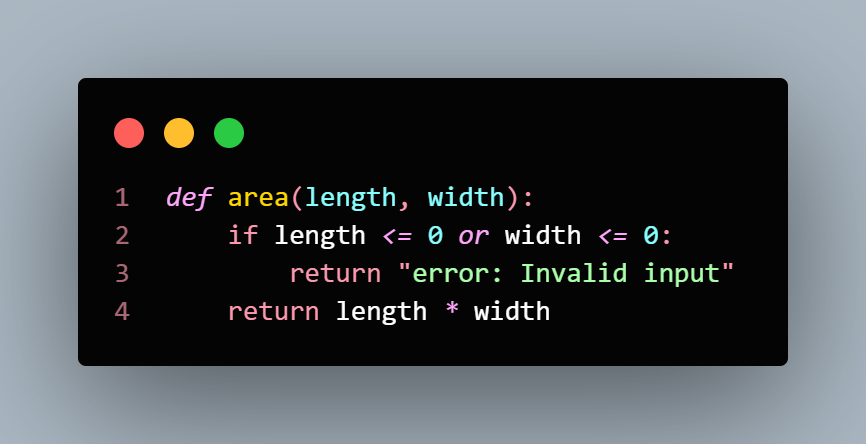
* Đường đi 1: A -> B -> C -> D -> F -> G
* Đường đi 2: A -> B -> C -> E -> G

# **Thiết lập các ca kiểm thử**

Thiết lập các dữ liệu đầu vào và kết quả mong muốn cho mỗi trường hợp kiểm thử. Với hai đường đi đã chọn, ta có thể thiết lập các dữ liệu vào và kết quả mong muốn như sau:

* Trường hợp kiểm thử 1: Dữ liệu vào là length = 10 và width = 5, kết quả mong muốn là area = 50.
* Trường hợp kiểm thử 2: Dữ liệu vào là length = -1 và width = 5, kết quả mong muốn là error massage = “Invalid input”.

# **Thực hiện code và kiểm thử**

1. Công nghệ sử dụng
   1. Ngôn ngữ lập trình: Python.
   2. Kiểm thử: sử dụng thư viện unittest của python.
2. Viết chương trình tính diện tích hình chữ nhật
3. Thực hiện các testcase bằng unittest

A computer screen shot of a program code

Description automatically generated

1. A black background with a black rectangle

   Description automatically generatedKết quả

# **Kết luận**

Chương trình tính toán diện tích của hình chữ nhật đã đáp ứng được các yêu cầu**.**

# **Bài tập**

1. Kiểm thử đơn vị một chương trình theo phương pháp kiểm thử dòng điều khiển với độ đo kiểm thử cho trước

Theo phương pháp kiểm thử dòng điều khiển, các bước nhằm kiểm thử một đơn vị chương trình với một độ đo kiểm thử cho trước là:

* Xây dựng đồ thị điều khiển cho đơn vị chương trình, trong đó các nút biểu diễn các khối xử lý, các điểm quyết định, các điểm nối và các điểm kết thúc. Các cạnh biểu diễn các luông điều khiển giữa các nút
* Chọn một độ đo kiểm thử phù hợp với mục tiêu và yêu cầu của kiểm thử. Có nhiều độ đo kiểm thử khác nhau như: bao phủ câu lệnh, bao phủ nhánh, bao phủ điều kiện, bao phủ dòng điều khiển, bao phủ dòng dữ liệu, bao phủ luồng dữ liệu v.v.
* Tìm các đường đi trong đồ thị điều khiển thỏa mãn độ đo kiểm thử đã chọn. Các đường đi này sẽ tạo thành các trường hợp kiểm thử cho đơn vị chương trình.
* Thiết lập các dữ liệu vào và kết quả mong muốn cho mỗi trường hợp kiểm thử. Các dữ liệu có thể được tạo ra dựa trên các tài liệu đặc tả kỹ thuật của chương trình hoặc sử dụng các công cụ sinh dữ liệu tự động.
* Thực hiện các trường hợp kiểm thử và so sánh kết quả thực tế với kết quả mong muốn. Nếu có sự khác biệt, ghi nhận và báo cáo lỗi để sửa chữa. Nếu không có sự khác biệt, xác nhận rằng đơn vị chương trình hoạt động đúng theo yêu cầu.

1. Các bài tập có trong slide bài giảng
   1. **Đồ thị dòng điều kiện**

A screenshot of a computer

Description automatically generated

Với độ phủ C1 (độ bao phủ câu lệnh), chúng ta cần kiểm tra mỗi câu lệnh ít nhất 1 lần. Dưới đây là các ca kiểm thử:

* Test case 1: score = -2 expected return ‘I’ ->(đường đi: 0-1-2)
* Test case 2: score = 10 expected return ‘A’ ->(đường đi: 0-1-3-4-12)
* Test Case 3: score = 8.5 expected res = ‘B’ ->(đường đi: 0-1-3-5-6-12)
* Test Case 4: score = 7.5 expected res = ‘C’ ->(đường đi: 0-1-3-5-7-8-12)
* Test Case 5: score = 6 expected res =”D’ -> (đường đi: 0-1-3-5-7-9-10-12)
* Test Case 6: score = 4 expected res = ‘F ->(đường đi: 0-1-3-5-7-9-11-12)

Với độ đo C2 (độ bao phủ nhánh), chúng ta cần kiểm tra mỗi nhánh ít nhất một lần. Dưới đây là các ca kiểm thử:

Để kiểm tra score < 0:

* + Ca kiểm thử 1: score = -1 để kiểm tra điều kiện score < 0 (đúng).
  + Ca kiểm thử 2: score = 0 để kiểm tra điều kiện score < 0 (sai).

Để kiểm tra score > 10:

* + Ca kiểm thử 3: score = 11 để kiểm tra điều kiện score > 10 (đúng).
  + Ca kiểm thử 4: score = 10 để kiểm tra điều kiện score > 10 (sai).

Để kiểm tra score >= 9:

* + Ca kiểm thử 5: score = 9 để kiểm tra điều kiện score >= 9 (đúng).
  + Ca kiểm thử 6: score = 8 để kiểm tra điều kiện score >= 9 (sai).

Để kiểm tra score >= 8:

* + Ca kiểm thử 7: score = 8 để kiểm tra điều kiện score >= 8 (đúng).
  + Ca kiểm thử 8: score = 7 để kiểm tra điều kiện score >= 8 (sai).

Để kiểm tra score >= 6.5:

* + Ca kiểm thử 9: score = 6.5 để kiểm tra điều kiện score >= 6.5 (đúng).
  + Ca kiểm thử 10: score = 6 để kiểm tra điều kiện score >= 6.5 (sai).

Để kiểm tra score >= 5:

* + Ca kiểm thử 11: score = 5 để kiểm tra điều kiện score >= 5 (đúng).
  + Ca kiểm thử 12: score = 4 để kiểm tra điều kiện score >= 5 (sai).
  1. **Đồ thị dòng điều kiện**

**A screenshot of a cell phone

Description automatically generated**

Độ phủ C1 (độ bao phủ câu lệnh): Mục tiêu là đảm bảo rằng mỗi câu lệnh trong chương trình được thực hiện ít nhất một lần.

* Ca kiểm thử 1: a = [0, 1, 2], n = 3. Kết quả mong đợi là 3. Đường đi (1-2-3-2-3-2-3-2-4)

Độ phủ C2 (độ bao phủ nhánh): Mục tiêu là đảm bảo rằng mỗi nhánh của chương trình được thực hiện ít nhất một lần.

* Ca kiểm thử 2: a = [1, 2, 3], n = 3. Kết quả mong đợi là 6. Điều này kiểm tra nhánh i<n. Đường đi (1-2-3-2-3-2-3-2-4)
* Ca kiểm thử 3: a = [], n = 0. Kết quả mong đợi là 0. Điều này kiểm tra nhánh i>=n.  Đường đi (1-2-4)

Kiểm thử vòng lặp for: Mục tiêu là kiểm tra các trường hợp khác nhau của vòng lặp.

* Ca kiểm thử 4: a = [], n = 0. Kết quả mong đợi là 0. Điều này kiểm tra trường hợp vòng lặp không được thực hiện. Đường đi (1-2-4)
* Ca kiểm thử 5: a = [1], n = 1. Kết quả mong đợi là 1. Điều này kiểm tra trường hợp vòng lặp được thực hiện một lần. Đường đi (1-2-3-2-4)
* Ca kiểm thử 6: a = [1, 2, 3], n = 3. Kết quả mong đợi là 6. Điều này kiểm tra trường hợp vòng lặp được thực hiện nhiều lần.