

VÕ TIẾN

Thảo luận kiến thức CNTT trường BK về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/211867931379013>



Nguyên Lí Ngôn Ngữ Lập Trình (PPL)

PPL2 - HK242

Giữa Kỳ

Thảo luận kiến thức CNTT trường BK
về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/211867931379013>

Mục lục

1	OOP	2
1.1	Giải MRO	2
1.1.1	Cách xác định MRO	2
1.2	Trắc Nghiệm	3
2	FP	7
2.1	Lý thuyết	7
2.2	Trắc Nghiệm	7
3	AST	11
3.1	Bài Tập	11
3.2	Trắc Nghiệm	12



1 OOP

1.1 Giải MRO

1. **Xác định lớp bắt đầu:** Đây là lớp mà bạn muốn tính MRO (thường là lớp cuối cùng trong hệ thống kế thừa).
2. **Liệt kê các lớp cha trực tiếp:** Ghi ra các lớp mà lớp hiện tại kế thừa, theo thứ tự từ trái sang phải trong khai báo.
3. **Tính MRO của từng lớp cha:** Đối với mỗi lớp cha, xác định MRO của nó (nếu nó cũng kế thừa từ lớp khác).
4. **Kết hợp tuần tự:**
 - Bắt đầu từ lớp hiện tại.
 - Lần lượt thêm các lớp từ danh sách cha và MRO của chúng, tuân theo hai quy tắc:
 - **Quy tắc 1:** Tôn trọng thứ tự khai báo (từ trái sang phải).
 - **Quy tắc 2:** Không thêm một lớp nếu nó đã xuất hiện trước đó, trừ khi tất cả các lớp khác trong danh sách hiện tại đã được thêm.

Ví dụ Trong [B, C], B đứng trước C. Vì vậy, ở bước đầu tiên, B được chọn trước C, dù C cũng là một lựa chọn hợp lệ. Điều này đảm bảo thứ tự khai báo của D(B, C) được giữ.

 - **Quy tắc 2:** Không thêm một lớp nếu nó đã xuất hiện trước đó, trừ khi tất cả các lớp khác trong danh sách hiện tại đã được thêm.

Ví dụ Ở bước 2, khi xem xét A và C, A không được thêm ngay vì nó nằm sau C trong [C, A, object]. Nếu thêm A trước C, sẽ vi phạm thứ tự của MRO(C). Do đó, C được ưu tiên thêm trước, và A chỉ được thêm khi không còn lớp nào khác chặn nó.
5. **Kết thúc với object:** Mọi lớp trong Python đều kế thừa từ object, nên nó luôn là lớp cuối cùng trong MRO.

1.1.1 Cách xác định MRO

Có 2 cách chính để xem MRO trong Python:

1. **Dùng thuộc tính `__mro__`:** Mỗi lớp trong Python có thuộc tính `__mro__` trả về một tuple chứa thứ tự phân giải phương thức.
2. **Dùng hàm `mro()`:** Gọi `ClassName.mro()` để lấy danh sách các lớp theo thứ tự MRO.

```
class X:
    pass

class Y:
    pass

class A(X, Y):
    pass

class B(Y):
    pass

class C(A, B):
    pass
```



Tính MRO của C:

1. **Cha trực tiếp của C:** [A, B].
2. **MRO của các lớp cha:**
 - $MRO(A) = [A, X, Y, object]$ (vì A kế thừa X, Y).



- $MRO(B) = [B, Y, object]$ (vì B kế thừa Y).

3. Merge:

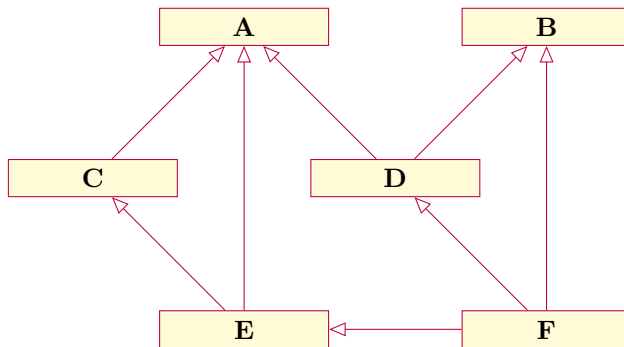
$MRO(C) = [C] + merge([A, X, Y, object], [B, Y, object], [A, B])$

- Lấy A, B, A -> Chọn A (vì A không ở giữa danh sách nào).
- Cập nhật: $[X, Y, object], [B, Y, object], [B]$.
- Lấy X, B, B -> Chọn B (vì X không ở đầu danh sách nào khác, nhưng B ưu tiên từ $[A, B]$).
- Cập nhật: $[X, Y, object], [Y, object], []$.
- Lấy X, Y -> Chọn X.
- Cập nhật: $[Y, object], [Y, object], []$.
- Lấy Y, Y -> Chọn Y.
- Cập nhật: $[object], [object], []$.
- Lấy object -> Chọn object.

4. **Kết quả:** $[C, A, B, X, Y, object]$.

1.2 Trắc Nghiệm

Cho class theo mô tả hình hãy làm 8 câu sau



- Theo ngôn ngữ lập trình hướng đối tượng kiểm tra kiểu tĩnh như Java với biến được khai báo kiểu là class A thì có thể gán giá trị của class nào sẽ đúng *các lớp con của nó*
 - A, B, C, D, E, F ~~X~~ *không có B*
 - A, C, D, E, F ~~X~~
 - A, B, D, E, F ~~X~~
 - C, D, E, F ~~X~~ *A?*
- Theo ngôn ngữ lập trình hướng đối tượng kiểm tra kiểu tĩnh như Java với biến được khai báo kiểu là class D thì có thể gán giá trị của class nào sẽ đúng
 - A, B, C, D, E, F *không có A*
 - A, B, F *vì B*
 - D, F
 - C, D, E, F
- Kết quả MRO của lớp F
 - F, D, E, C, A, B
 - F, D, E, C, B, A
 - F, D, E, A, B
 - F, D, E, C, A

Nếu python mỗi class đều có hàm sau (bỏ qua lỗi ở class cuối được gọi super)

```
def foo(self):
    super().foo()
```

- Nếu khai báo như sau kết quả của đoạn code



```
def foo(votien : E):  
    votien.foo()  
foo(F())
```

- a) F, D, E, C, A, B
- c) F, D, E, A, B

- b) F, D, E, C, B, A
- d) error

5. Nếu khai báo như sau kết quả của đoạn code

```
def foo(votien : E):  
    votien.foo()  
foo(E())
```

- a) F, D, E, C, A, B
- c) E, C, A

- b) E, A, C
- d) error

6. Nếu khai báo như sau kết quả của đoạn code

```
def foo(votien : E):  
    votien.foo()  
foo(A())
```

- a) F, D, E, C, A, B
- c) A

- b) E, A, C
- d) error

7. Nếu khai báo đoạn code F().foo() và Class D không có hàm foo(), kết quả

- a) F, E, C, A, B
- c) F, E, A, B

- b) F, E, C, B, A
- d) error

8. Nếu khai báo đoạn code F().foo() và Class D với foo() không gọi super().foo(), kết quả

- a) F, D
- c) F, E, A, B

- b) F, D, E, C, B, A
- d) error

Áp dụng mô tả sau cho 2 câu sau:

Cho một đoạn mã trong ngôn ngữ Python như sau:

```
class Shape:  
    def area(self):  
        return 0  
  
class Square(Shape):  
    def __init__(self, s):  
        self.s = s  
  
    def area(self):  
        return self.s ** 2  
  
class Circle(Shape):  
    def __init__(self, r):  
        self.r = r  
  
    def area(self):  
        return 3.14 * self.r ** 2  
  
def calculate_total_area(shapes):  
    total_area = 0  
    for shape in shapes:  
        total_area += shape.area()  
    return total_area
```

Square và Circle ghi đè
method area của shape
⇒ đa hình

do ghi đè ⇒ run time mới xác
định được (trong hàm cuối)
⇒

Sub typing polymorphism

→ Khi chạy mới biết
shape.area() của ai



```
square, circle = Square(5), Circle(3)
print(calculate_total_area([square, circle]))
```

9. Trong ngữ cảnh của đoạn mã trên, tính chất nào của lập trình hướng đối tượng được thể hiện mạnh mẽ nhất? *→ có nhưng không mạnh bằng → ko có đư thừa kô*

- a) Single inheritance b) Multiple inheritance
c) Parametric Polymorphism ~~d) Subtyping polymorphism~~

10. Kết quả in ra màn hình của đoạn mã trên là:

- Định tham số* *→ chỉ mang template* *→ $\pi \cdot 3^2 + 5^2$*
a) 53.26 b) 43.84 c) 28.26 d) 34.0

11. Trong ngữ cảnh của đoạn mã dưới c++, tính chất nào của lập trình hướng đối tượng được thể hiện mạnh mẽ nhất?

```
template <typename T>
class Container {
private:
    T value;
public:
    Container(T v) : value(v) {}
    T getValue() { return value; }
};

int main() {
    Container<int> intContainer(42);
    Container<string> strContainer("Hello");

    cout << intContainer.getValue() << endl; // 42
    cout << strContainer.getValue() << endl; // Hello

    return 0;
}
```

ko có kô thừa ⇒ Low A

⇒ B

Template ⇒ C

- a) Single inheritance b) Multiple inheritance
~~c) Parametric Polymorphism~~ d) Subtyping polymorphism

Áp dụng mô tả sau cho 2 câu sau

Cho một đoạn mã trong ngôn ngữ Python như sau:

```
class A:
    def show(self): return "A"

class B(A):
    def show(self): return "B"

class C(A):
    def show(self): return self.__class__.__name__

class D(C, B): pass
class E(B, C): pass
class F(E, B): pass
```

12. Thứ tự phân giải phương thức (MRO) của F là:

- a) F, D, E, B, C, object b) F, E, B, C, A, object
c) F, E, D, B, A, object d) F, E, C, B, D, object

13. Để kết quả trả về của F().show() là "F" thì cần thực hiện gì sau đây?

- a) Đoạn mã trên đã cho kết quả F().show() là "F"



- b) Thêm lớp cơ sở C vào cuối danh sách hiện tại của lớp F
- c) Xóa đi lớp cơ sở B trong lớp F
- d) Thay đổi thứ tự lớp cơ sở của E thành C, B

14. Giả sử mã nguồn của file `lib.py` như sau:

```
class A:
    def f(self):
        print('f in A')

class B(A):
    def g(self):
        super().f()
```

Để dùng được phương thức `g()` của lớp B nhưng thay đổi chức năng của `f()`, một lớp C với phương thức `f` mới và một lớp D mới được viết trên Python như sau:

```
from lib import *

class C(__):
    def f(self):
        print('f in C')

class D(__):
    pass
```

`D().g()`

Hãy cho biết cần phải khai báo lớp cha của C và D như thế nào để dòng lệnh `D().g()` sẽ có kết quả là 'f in C'?

- a) `class C(B):` và `class D(A, C):`
 - b) `class C(B):` và `class D(C, A):`
 - c) `class C(A):` và `class D(C, B):`
 - d) `class C(A):` và `class D(B, C):`
15. Cho các khai báo sau được viết trên một ngôn ngữ lập trình hướng đối tượng dùng kiểm tra kiểu tĩnh như **Scala**:

```
class A { def foo() = print("a") }
class B extends A { } // B is a subclass of A
class C extends A { override def foo() = print("c") } // C is a subclass of A
class D extends B { override def foo() = print("d") } // D is a subclass of B
```

Biết rằng, biến `b` được khai báo kiểu B và đang tham chiếu đến một đối tượng nào đó. Cho một số nhận định về kết quả được in ra khi gọi `b.foo()`:

- (a) c (nếu `b` đang tham chiếu đến một đối tượng C) X ko đc
- (b) d (nếu `b` đang tham chiếu đến một đối tượng D) ✓
- (c) a (nếu `b` đang tham chiếu đến một đối tượng A) X
- (d) a (nếu `b` đang tham chiếu đến một đối tượng B) ✓

Các nhận định đúng là:

- a) (a), (b), (c) và (d) ✗
- b) (b) và (d) ✓
- c) (c) và (d) ✗
- d) (b) và (c) ✗



16. Khái niệm "decorator" trong Python có thể được sử dụng để làm gì?

- ✗ Thay đổi hoặc mở rộng hành vi của một hàm "gói" một hàm = một hàm khác
- b) Thêm chức năng mới vào một hàm
- c) Định nghĩa một hàm mới từ một hàm thư viện khác
- d) Xóa chức năng được chỉ định cụ thể từ một hàm



2 FP

2.1 Lý thuyết

map(), filter(), và reduce()

- **map(function, iterable)**: Áp dụng hàm lên từng phần tử của iterable.
- **filter(function, iterable)**: Lọc các phần tử thỏa mãn điều kiện.
- **reduce(function, iterable)**: Gộp iterable thành một giá trị duy nhất (từ functools).

```
numbers = [1, 2, 3, 4]
```

```
# map: Bình phương từng số
squared = list(map(lambda x: x**2, numbers)) # [1, 4, 9, 16]
```

```
# filter: Lọc số chẵn
evens = list(filter(lambda x: x % 2 == 0, numbers)) # [2, 4]
```

```
# reduce: Tính tổng
sum_all = reduce(lambda x, y: x + y, numbers) # 10
```

List Comprehension và Generator Expression

- Thay thế map() và filter() bằng cú pháp ngắn gọn hơn.
- Generator expression tiết kiệm bộ nhớ bằng cách không tạo toàn bộ danh sách cùng lúc.

```
# List comprehension
squared = [x**2 for x in range(5)] # [0, 1, 4, 9, 16]
```

```
# Generator expression
squared_gen = (x**2 for x in range(5))
print(next(squared_gen)) # 0
print(next(squared_gen)) # 1
```

Currying Currying là kỹ thuật biến một hàm nhiều tham số thành chuỗi các hàm một tham số.

```
def add(a, b):
    return a + b
```

```
add_five = partial(add, 5) # Gắn a = 5
print(add_five(3)) # Kết quả:
```

list (1+4) (2+5) (3+6)

2.2 Trắc Nghiệm

1. kết quả đoạn code sau

```
list(map(lambda x, y: x + y, [1, 2, 3], [4, 5, 6]))
```

~~a) [5, 7, 9]~~

b) [1, 2, 3, 4, 5, 6]

c) [4, 5, 6]

d) *Error*

2. kết quả đoạn code sau

```
def is_even(n):
    return n % 2 == 0
```

```
list(filter(is_even, range(10)))
```

~~a) [0, 2, 4, 6, 8]~~

b) [1, 3, 5, 7, 9]

c) []

d) [2, 4, 6, 8]

3. kết quả đoạn code sau



```
def multiply(x, y):
    return x * y
```

(10) * 2 = 20

```
reduce(multiply, [1, 2, 3, 4], 10)
```

a) 240

b) 0

c) 10

d) 720

4. kết quả đoạn code sau

```
def outer(x):
    def inner(y):
        return x + y
    return inner
```

```
f = outer(10)
result = f(f(5))
```

= inner(y) = y + 10

f(5) = 15

f(15) = 25

a) 15

b) 20

d) 30

5. kết quả đoạn code sau

```
def make_closure():
    x = 5
    def closure():
        return x
    x = 10
    return closure
```

→ return 5

→ return 5

```
closure_func = make_closure()
closure_func()
```

a) 5

b) 10

c) Error

d) None

6. kết quả đoạn code sau

```
def square(x): return x ** 2
def double(x): return x * 2
numbers = [1, 2, 3, 4, 5]
result = map(square, filter(lambda x: x % 2 == 0, map(double, numbers)))
print(list(result))
```

[2, 4, 6, 8, 10]

[2, 4, 6, 8, 10]

a) [4, 16]

b) [4, 16, 36, 64, 100]

c) [1, 4, 9, 16, 25]

d) [2, 4, 6, 8, 10]

7. kết quả đoạn code sau

```
def square(x): return x ** 2
def double(x): return x * 2
numbers = [1, 2, 3, 4, 5]
result = map(square, filter(lambda x: x % 2 == 0, map(double, numbers)))
print(list(result))
```

a) [4, 16]

b) [4, 16, 36, 64, 100]

c) [1, 4, 9, 16, 25]

d) [2, 4, 6, 8, 10]

8. Cho đoạn mã trong Python như sau:

```
def foo(f, x): return f(x)
foo(lambda a: a ** 2, 4)
```

f(x) · x = x² = 16

Đoạn mã nào trong ngôn ngữ lập trình C++ sau đây tương đương với đoạn mã Python trên:

```
(a) int foo(int (*f)(int), int x) { return f(x); }
int main() {
    auto lambda = int f(int a) { return a * a; };
    foo(lambda, 4);
```

* ko khai báo đng hàm lambda



() (int) máy*

```

    return 0;
}
(b) int foo(int *f(int), int x) { return f(x); }
    int main() {
        auto lambda = int f(int a) { return a * a; };
        foo(lambda, 4);
        return 0;
    }

```

? ~~(c)~~ int foo(int (*f)(int), int x) { return f(x); }

```

    int main() {
        auto lambda = [] (int a) { return a * a; };
        foo(lambda, 4);
        return 0;
    }

```

```

(d) int foo(int *f(int), int x) { return f(x); }
    int f(int a) { return a * a; }
    int main() {
        foo(f, 4);
        return 0;
    }

```

9. Có bao nhiêu hàm bậc cao (high-order function) trong số các hàm thư viện sau: max, filter, round, abs, map?

a) 1

b) 2

c) 3

d) 4

hàm bậc cao -> Nhận hoặc trả về một hàm

10. Cho hàm is_matrix được định nghĩa trong Python như sau:

```

def is_matrix(matrix):
    if not matrix: return False
    fl = len(matrix[0])
    return reduce(_____, matrix, True)

```

True là giá trị tích lũy, sẽ là false nếu hàm reduce lặp 1 lần

Đoạn mã nào cần điền vào khoảng trống ở dòng 4 để hàm trên trả về True khi matrix là một ma trận, ngược lại trả về False?

a) lambda x, y: x or len(y) == fl
 c) lambda x, y: x and len(y) == fl

Lặp qua từng dòng của matrix, xác định len có bằng hàng đầu
 không b) lambda x, y: fl and len(y) == x
 d) lambda x, y: x and len(y) == len(matrix)

x là true || false

11. Các ngôn ngữ lập trình hàm thuần khiết không có các cấu trúc lặp dựa trên biểu thức luận lý như phát biểu while, do while trên C vì trên ngôn ngữ lập trình hàm:

immutability

a) không thể thay đổi giá trị của biến
 c) có hàm bậc cao thay thế

b) không có biểu thức luận lý
 d) dùng đệ quy để thay thế

12. Sau khi hoàn thành (lập trình và kiểm thử) hàm funcA viết trên Python để thực hiện một chức năng trên web, bạn muốn điều chỉnh để chức năng này chỉ được sử dụng sau khi người dùng đăng nhập. Một giải pháp đơn giản là thêm @login_required vào trước dòng khai báo hàm funcA. Trên Python, login_required là:

a) một tổ chức dữ liệu
 c) một cấu trúc điều khiển

là một decorator, và decorator trong Python được thực hiện bằng cách sử dụng hàm bậc cao
 b) một lớp trong thư viện
 d) một hàm bậc cao

13. Hãy cho biết kết quả xuất ra màn hình của đoạn mã (trong ngôn ngữ Python) sau:

```

def square(x): return x ** 2
def double(x): return x * 2
numbers = [1, 2, 3, 4, 5]

```



```

[8, 32] [4, 16] [1, 4, 9, 16, 25]
result = map(double, filter(lambda x: x % 2 == 0, map(square, numbers)))
print(list(result))

```

- a) [4, 16] b) [4, 16, 36, 64, 100] c) [2, 18, 50] d) [8, 32]

14. Hàm hợp của các hàm f_1, f_2, \dots, f_n là hàm h sao cho $h(x) = f_n(\dots(f_2(f_1(x))))$. Hãy điền vào chỗ trống trong đoạn mã sau để hàm compose sẽ trả về hàm hợp của các thông số của nó?

```

from functools import reduce
def compose(*f):

```

```

    def inner(x):

```

```

        return reduce(lambda a, b: _____, x)

```

```

    return inner

```

- a) a(b), f b) b(a), f[::-1] c) b(a), f d) a(b), f[::-1]

Nếu ta muốn $h(x) = f_n(\dots(f_2(f_1(x))))$, thì `reduce` cần áp dụng các hàm từ trái sang phải, bắt đầu từ `x`, rồi áp dụng f_1, f_2, \dots , đến f_n .

`reduce(function, iterable, initializer)`

`b` là iterable, nên nó sẽ lần lượt là f_1, f_2, \dots

Bắt đầu: $a = x, b = f_1 \rightarrow f_1(x)$

15. Xét đoạn mã sau trong Python:

```

from functools import reduce
def mystery_function(lst):

```

```

    return reduce(lambda x, y: x + [y] if y % 2 == 0 else x, lst, [])

```

```

print(mystery_function([1, 2, 3, 4, 5, 6]))

```

Kết quả của đoạn mã trên là gì?

- a) [1, 2, 3, 4, 5, 6] b) [2, 4, 6]
c) [[2], [4], [6]] d) [1, 3, 5]

16. Xét đoạn mã Python sau:

```

def make_multiplier(n):
    return lambda x: x * n

```

```

mult_by_3 = make_multiplier(3)

```

```

result = list(map(mult_by_3, filter(lambda x: x % 3 == 0, range(10))))
print(result)

```

Giá trị của `result` là gì?

- a) [0, 3, 6, 9] b) [0, 9, 18, 27] c) [3, 6, 9] d) [0, 6, 12, 18]

17. Xét đoạn mã Python sau:

```

def create_functions():
    return [lambda x: x + i for i in range(5)]

```

```

functions = create_functions()
result = [f(10) for f in functions]
print(result)

```

Giá trị của `result` là gì?

- a) [10, 11, 12, 13, 14] b) [14, 14, 14, 14, 14]
c) [10, 11, 12, 13, 14, 15] d) [15, 15, 15, 15, 15]

Hàm này định nghĩa một danh sách các hàm lambda bằng cách sử dụng list comprehension: `[lambda x: x + i for i in range(5)]`. Vòng lặp `for i in range(5)` dự định tạo ra 5 hàm lambda, mỗi hàm sẽ cộng một giá trị khác nhau của `i` (từ 0 đến 4) vào đối số `x`. Tuy nhiên, do late binding, biến `i` không được liên kết với giá trị của nó tại thời điểm hàm lambda được định nghĩa. Thay vào đó, `i` được tham chiếu đến trong phạm vi bao ngoài khi hàm lambda được gọi.

Dòng này sử dụng list comprehension để gọi từng hàm `f` trong danh sách `functions` với đối số 10 và thu thập kết quả vào danh sách `result`.

Khi các hàm lambda được gọi ở đây, vòng lặp `for i in range(5)` trong `create_functions()` đã hoàn thành. Giá trị cuối cùng của `i` sau vòng lặp là 4.

Do late binding, tất cả các hàm lambda trong danh sách `functions` đều tham chiếu đến cùng một biến `i` trong phạm vi bao ngoài, và giá trị của `i` lúc này là 4.

Vì vậy, mỗi hàm lambda thực tế trở thành `lambda x: x + 4`.



3 AST

3.1 Bài Tập

Cho biểu thức antlr sau hãy trả lời các câu hỏi sau

```
// parser
expression: expression1 CONCAT expression1 | expression1;
expression1: expression2 (NE | LT | LE | GT | GE) expression2 | expression2;
expression2: expression3 (AND | OR) expression2 | expression3;
expression3: expression3 (ADD | SUB) expression4 | expression4;
expression4: expression5 (MUL | DIV) expression4 | expression5;
expression5: NOT expression5 | expression6;
expression6: SUB expression7 | expression7;
expression7: ID | INTLIT;
```

Code python AST

```
class BinExpr(Expr):
    def __init__(self, op: str, left: Expr, right: Expr):
        self.op = op
        self.left = left
        self.right = right

class UnExpr(Expr):
    def __init__(self, op: str, val: Expr):
        self.op = op
        self.val = val

class Id(LHS):
    def __init__(self, name: str):
        self.name = name

class IntegerLit(Expr):
    def __init__(self, val: int):
        self.val = val
```

1. Vẽ cây parser, cây AST, biểu thức AST, Tính toán giá trị nếu được $1+2*3+2*4*3$
2. Vẽ cây parser, cây AST, biểu thức AST, Tính toán giá trị nếu được $1 + 1 > 2 * 4$
3. Vẽ cây parser, cây AST, biểu thức AST, Tính toán giá trị nếu được $1 \&\& 2 \&\& 3 * 4$
4. Vẽ cây parser, cây AST, biểu thức AST, Tính toán giá trị nếu được $1 < 2 < 3$
5. Vẽ cây parser, cây AST, biểu thức AST, Tính toán giá trị nếu được $1 < 3 \&\& 1 < 2$
6. Vẽ cây parser, cây AST, biểu thức AST, Tính toán giá trị nếu được $3*16/2*4$
7. Vẽ cây parser, cây AST, biểu thức AST, Tính toán giá trị nếu được $-2 + !1$
8. Vẽ cây parser, cây AST, biểu thức AST, Tính toán giá trị nếu được $a + b \text{ concat } c + d$



3.2 Trắc Nghiệm

Cây parser đúng không có nút riêng cho giá trị cụ thể như "abc" dưới nút ID. Thay vào đó, giá trị "abc" được xem như là thông tin hoặc thuộc tính gắn liền với nút ID.

1. Chiều cao tối đa của cây parser sau

```
ID: [a-z]+;
WS: [ \t\n] -> skip;
```

```
program: expr+ EOF;
expr: ID;
```

- a) 1 b) 2 **c) 3** d) 4

2. Chuyển đổi thành cây AST cho đoạn code c++ kết quả của phần Type, hãy chọn cây hợp lí nhất

```
votien a[2][3];
```

và các lớp AST được khai báo trong ngôn ngữ Python3 như sau:

```
class AST
class Type(AST):
class ArrayType(Type): # dims:List[Expr], eleType:Type
class IntLiteral(Expr): # value:int
class Id(Type, Expr): # name : str

a) ArrayType([IntLiteral(2), IntLiteral(3)], Id("votien"))
b) ArrayType(Id("votien"), [IntLiteral(2), IntLiteral(3)])
c) ArrayType([IntLiteral(3), IntLiteral(2)], Id("votien"))
d) ArrayType([IntLiteral(3)], Id("votien"))
```

Làm 7 câu sau với Ngôn ngữ dưới đây

Cho đoạn ngữ pháp được viết trong ANTLR4 cho ngôn ngữ VOTIEN như sau:

```
ID: [a-z]+;
LB: '[';
RB: ']';
WS: [ \t\n] -> skip;
INTLIT: [0-9]+;
```

```
program: expr+ EOF;
expr: ID (LB INTLIT RB)*;
```

và các lớp AST được khai báo trong ngôn ngữ Python3 như sau:

```
class AST
class Program(AST): # expr: List[Expr]
class Expr(AST):
class ArrayCell(Expr): # arr:Expr, idx:List[int]
class Id(Expr): # name : str
```

Đoạn mã sau với một số chỗ trống và một số đoạn bị che (Hidden code) được sử dụng để thực hiện việc sinh AST từ câu cú pháp được tạo ra bởi bộ phân tích cú pháp do ANTLR sinh ra trên Python3 là

```
1 class ASTGenerator(VOTIENVisitor):
2     def visitProgram(self, ctx):
3         return _____(1)_____
4     def visitExpr(self, ctx):
5         if _____(2)_____ return _____(3)_____
6         return ArrayCell(_____ (4)_____)
```

Trong phương thức visitExpr, chúng ta cần phân biệt giữa nút Id và ArrayCell

3. Chỗ trống (1) ở dòng số 3 nên là

- a) Program(list(map(self.visit, ctx.expr())))
b) Program(filter(map(self.visit, ctx.expr())))



- ```
class AST
class Program(AST): # stmts: List[Stmt]
class Stmt(AST)
```



```

class Assign Stmt: # lhs: Id, right: Exp
class Exp AST
class BinExp Exp: # op: str, left: Exp, right: Exp
class IdxOp Exp: # base: Exp, idx: int
class IntLit Exp: # value: int
class Id Exp: # name: str

```

Với chuỗi nhập viết trên ngôn ngữ BMirror như sau:

```

a, b, c = 1 + 2, a * 2, 4 / b
d = arr[1][2][3]

```

AST tương ứng cần phải được sinh ra như sau:

```

Program([
 Assign(Id("a"), BinExp("+", IntLit(1), IntLit(2))),
 Assign(Id("b"), BinExp("*", Id("a"), IntLit(2))),
 Assign(Id("c"), BinExp("/", IntLit(4), Id("b"))),
 Assign(Id("d"), IdxOp(IdxOp(IdxOp(Id("arr"), 1), 2), 3))
])

```

Đoạn mã sau với một số chỗ trống và một số đoạn bị che ( Hidden code) được sử dụng để thực hiện việc sinh AST từ câu cú pháp được tạo ra bởi bộ phân tích cú pháp do ANTLR sinh ra trên Python3:

```

1 class ASTGenerator(BMirrorVisitor):
2 def visitProgram(self, ctx):
3 assign_stmts = []
4 for assign in ctx.assign_stmt():
5 assign_stmts += self.visit(assign)
6 return Program(assign_stmts)
7
8 def visitAssign_stmt(self, ctx):
9 ids, explist = self.visit(ctx.assign())
10 return _____(1)_____
11
12 def visitAssign(self, ctx):
13 if ctx.getChildCount() == 3:
14 return _____(2)_____
15 ids, explist = self.visit(ctx.assign())
16 return [Id(ctx.ID().getText())] + _____(3)_____ + [self.visit(ctx.exp())]
17
18 def visitExp(self, ctx):
19 ## Hidden Code
20
21 def visitTerm(self, ctx):
22 ## Hidden Code
23
24 def visitFact(self, ctx):
25 if ctx.ID():
26 return Id(ctx.ID().getText())
27 elif ctx.INTLIT():
28 return IntLit(int(ctx.INTLIT().getText()))
29 return _____(4)_____
30
31 def visitIdx_op(self, ctx):
32 return reduce(lambda x, y: _____(5)_____, _____(6)_____, _____(7)_____)

```

10. Kiểu đầy đủ của tham số ctx trong phương thức visitAssign\_stmt là

- a) BMirrorParser.Assign\_stmtContext      b) BMirrorParserTree.Assign\_stmtContext



- c) ~~BMirror~~.Assign\_stmtContext                      d) BMirrorParser.Assign\_stmt
11. Chỗ trống (7) ở dòng số 23 nên là  
 a) ,Id(ctx.ID().getText())  
 c) ,ctx.ID()
12. Chỗ trống (6) ở dòng số 23 nên là  
 a) ctx.ID()  
 c) self.visit(ctx.INTLIT())  
 b) int(ctx.INTLIT().getText())  
 d) ctx.INTLIT() **literal** -> phải là một list
13. Chỗ trống (5) ở dòng số 23 nên là  
 a) IdxOp(y,x)  
 c) IdxOp(x,int(y.getText()))  
 b) IdxOp(x,int(y)) **x nhận giá trị init = []**  
 d) IdxOp(y,int(x))
14. Chỗ trống (4) ở dòng số 21 nên là  
 a) [self.visit(ctx.idx\_op())]  
 c) self.visit(ctx.idx\_op())  
 b) IdxOp(self.visit(ctx.idx\_op()))  
 d) ctx.idx\_op()
15. Nhận định nào sau đây là đúng?  
 a) Số loại đối tượng mà visitExp và visitFact có thể trả về là giống nhau và là 4.  
 b) visitFact có thể trả về đối tượng Id trong khi visitTerm thì chỉ có thể trả về đối tượng BinExp. **hoặc đối tượng term**  
 c) visitFact có thể trả về đối tượng IntLit trong khi visitExp thì chỉ có thể trả về đối tượng Exp. term  
 d) Số loại đối tượng mà visitExp và visitTerm có thể trả về là 1 và là BinExp. **có thể không phải là BinExp**
16. Chỗ trống (2) ở dòng số 11 nên là  
 a) Assign(Id(ctx.ID().getText()), self.visit(ctx.exp()))  
 c) Assign(Id(ctx.ID().getText()), ctx.exp())  
 b) ~~Id~~(ctx.ID().getText()) , [self.visit(ctx.exp())]  
 d) (Id(ctx.ID().getText()), self.visit(ctx.exp()))
17. Chỗ trống (3) ở dòng số 13 nên là  
 a) self.visit(ctx.assign())  
 c) ids+explist  
 b) ids,explist  
 d) ctx.assign()
18. Biết rằng zip là hàm để tạo một danh sách các phần tử tuple được lấy từ các phần tử có chỉ số tương ứng của các danh sách đầu vào. Chỗ trống (1) ở dòng số 8 nên là  
 a) [Assign(idc, exp) for IDC, exp in list(zip(ids[:-1], explist))]  
 b) [Assign(idc, exp) for IDC, exp in list(zip(ids[:-1], explist[:-1]))]  
 c) [Assign(idc, exp) for IDC, exp in list(zip(ids, explist[:-1]))]  
 d) [Assign(idc, exp) for IDC, exp in list(zip(ids, explist))]
19. Chọn lệnh phù hợp để thay thế các dòng lệnh từ 3 đến 5?  
 a) return Program(reduce(lambda prev, curr: prev + self.visit(curr), ctx.assign\_stmt()))  
 b) return Program(reduce(lambda prev, curr: prev + [self.visit(curr)], ctx.assign\_stmt()))  
 c) return Program(reduce(lambda prev, curr: prev + self.visit(curr), ctx.assign\_stmt(), []))  
 d) return Program(reduce(lambda prev, curr: prev + [self.visit(curr)], ctx.assign\_stmt(), []))
- Program([  
 Assign(Id("a"), BinExp("+", IntLit(1), IntLit(2))),  
 Assign(Id("b"), BinExp("\*", Id("a"), IntLit(2))),  
 Assign(Id("c"), BinExp("/", IntLit(4), Id("b"))),  
 Assign(Id("d"), IdxOp(IdxOp(IdxOp(Id("arr"), 1), 2), 3))  
 ])