

Trang của tôi / Khoá học / Học kỳ I năm học 2021-2022 (Semester 1 - Academic year 2021-2022)

- / <u>Đại Học Chính Qui (Bacherlor program (Full-time study))</u>
- / Khoa Khoa học và Kỹ thuật Máy tính (Faculty of Computer Science and Engineering.) / Khoa Học Máy Tính
- / Nguyên lý ngôn ngữ lập trình (CO3005) Nguyễn Hứa Phùng (DH_HK211) / 4-OOP / OOP Programming

Đã bắt đầu vào	Tuesday, 7 September 2021, 7:52 AM
lúc	
Tình trạng	Đã hoàn thành
Hoàn thành vào	Monday, 13 September 2021, 5:00 PM
lúc	
Thời gian thực	6 ngày 9 giờ
hiện	
Điểm	3,00/3,00
Điểm	10,00 của 10,00 (100 %)

```
Câu hồi 1
Chính xác
Điểm 1,00 của 1,00
```

To express an arithmetic expression, there are 5 following classes:

Exp: general arithmetic expression

BinExp: an arithmetic expression that contains one binary operators (+,-,*,/) and two operands

UnExp: an arithmetic expression that contains one unary operator (+,-) and one operand

IntLit: an arithmetic expression that contains one integer number

FloatLit: an arithmetic expression that contains one floating point number

Define these classes in Python (their parents, attributes, methods) such that their objects can response to eval() message by returning the value of the expression. For example, let object x express the arithmetic expression 3 + 4 * 2.0, x.eval() must return 11.0

Answer: (penalty regime: 0 %)

```
class Exp:
 2 🔻
 3 ▼
        def eval():
 4
            pass
 5
 6
 7 ▼ class UnExp:
 8 •
        def __init__(self, operator, arg):
 9
            self.operator = operator
10
            self.arg = arg
11
12 🔻
        def eval(self):
            if self.operator == '+':
13 v
                return self.arg.value
14
            if self.operator == '-':
15 ▼
16
                return -self.arg.value
17
18
    class BinExp():
19 🔻
20 🔻
        def __init__(self, left, operator, right):
            self.operator = operator
21
22
            self.left = left
23
            self.right = right
```

	Test	Expected	Got	
~	<pre>print(x1.eval())</pre>	1	1	~
~	<pre>print(x2.eval())</pre>	2.0	2.0	~
~	<pre>print(x3.eval())</pre>	2	2	~
~	<pre>print(x4.eval())</pre>	-1	-1	~
~	<pre>print(x5.eval())</pre>	7.0	7.0	~

Passed all tests!

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

```
Câu hởi 2
Chính xác
Điểm 1,00 của 1,00
```

Extend the contents of classes Exp, BinExp, UnExp, IntLit, FloatLit such that they can response to printPrefix() message to return the string corresponding to the expression in prefix format. Note that, unary operator +/- is printed as +./-. in prefix format and there is a space after each operator or operand. For example, when receiving message printPrefix(), the object expressing the expression -4 + 3 * 2 will return the string "+ -, 4 * 3 2 "

Answer: (penalty regime: 0 %)

```
1
 2 🔻
     class Exp:
 3 ▼
        def eval():
 4
            pass
 6
 7 🔻
    class UnExp:
        def __init__(self, operator, arg):
 8 •
 9
            self.operator = operator
10
            self.arg = arg
11
        def eval(self):
12 🔻
13 🔻
            if self.operator == '+':
                return self.arg.value
14
15
            if self.operator == '-':
                return -self.arg.value
16
17
        def printPrefix(self):
18 🔻
            return self.operator + '. '+self.arg.printPrefix()
19
20
21
22 ▼ class BinExp():
        def __init__(self, left, operator, right):
23 ▼
```

	Test	Expected	Got	
~	<pre>print(x1.printPrefix())</pre>	1	1	~
~	<pre>print(x2.printPrefix())</pre>	2.0	2.0	~
~	<pre>print(x3.printPrefix())</pre>	+ 1 1	+ 1 1	~
~	<pre>print(x4.printPrefix())</pre>	1	1	~
~	<pre>print(x5.printPrefix())</pre>	+ 1 * 4 2.0	+ 1 * 4 2.0	~

Passed all tests! 🗸

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

```
Câu hỏi {f 3}
Chính xác
Điểm 1,00 của 1,00
```

As in the previous question, when a task is added into expression classes, new methods are added into these classes. Please change the way these classes are implemented in such a way that these classes do not change their contents when new tasks are added into these classes:

- Define class Eval to calculate the value of an expression
- Define class PrintPrefix to return the string corresponding to the expression in prefix format
- Define class PrintPostfix to return the string corresponding to the expression in postfix format

Let x be an object expressing an expression, x.accept(Eval()) will return the value of the expression x, x.accept(PrintPrefix()) will return the expression in prefix format and x.accept(PrintPostfix()) will return the expression in postfix format.

Be careful that you are not allowed to use type(), isinstance() when implementing this exercise

Tip: Use Visitor pattern.

Answer: (penalty regime: 0 %)

```
class Exp:
 2 •
        def accept(self, visitor):
 3 ,
 4
            return visitor.visit(self)
 5
 6
    class UnExp(Exp):
 7 🔻
 8 🔻
        def __init__(self, operator, arg):
9
            self.operator = operator
10
            self.arg = arg
11
12 🔻
        def eval(self):
13 🔻
            if self.operator == '+':
14
                return self.arg.value
            if self.operator == '-':
15 ▼
16
                return -self.arg.value
17
        def printPrefix(self):
18 v
19
            return self.operator + '. '+self.arg.printPrefix()
20
21 🔻
        def printPostfix(self):
            return self.arg.printPrefix()+' '+self.operator + '.'
22
23
```

	Test	Expected	Got	
~	<pre>print(x1.accept(Eval()))</pre>	1	1	~
	<pre>print(x1.accept(PrintPrefix()))</pre>	1	1	
	<pre>print(x1.accept(PrintPostfix()))</pre>	1	1	
~	<pre>print(x2.accept(Eval()))</pre>	2.0	2.0	~
	<pre>print(x2.accept(PrintPrefix()))</pre>	2.0	2.0	
	<pre>print(x2.accept(PrintPostfix()))</pre>	2.0	2.0	
~	<pre>print(x3.accept(Eval()))</pre>	2	2	~
	<pre>print(x3.accept(PrintPrefix()))</pre>	+ 1 1	+ 1 1	
	<pre>print(x3.accept(PrintPostfix()))</pre>	1 1 +	1 1 +	
~	<pre>print(x4.accept(Eval()))</pre>	-1	-1	~
	<pre>print(x4.accept(PrintPrefix()))</pre>	1	1	
	<pre>print(x4.accept(PrintPostfix()))</pre>	1	1	
~	<pre>print(x5.accept(Eval()))</pre>	7.0	7.0	~
	<pre>print(x5.accept(PrintPrefix()))</pre>	+ 1 * 4 2.0	+ 1 * 4 2.0	
	<pre>print(x5.accept(PrintPostfix()))</pre>	1 4 2.0 * +	1 4 2.0 * +	

Passed all tests! 🗸

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

■ OOP Quiz

Chuyển tới...

Link Video buổi học 7/9/2021 ▶

Copyright 2007-2021 Trường Đại Học Bách Khoa - ĐHQG Tp.HCM. All Rights Reserved. Địa chỉ: Nhà A1- 268 Lý Thường Kiệt, Phường 14, Quận 10, Tp.HCM. Email: elearning@hcmut.edu.vn Phát triển dựa trên hệ thống Moodle