-Построить ленивое дерево выражений, позволяющее эффективно вычислять сложные математические выражения.

**class** Node:

**def** evaluate(*self*):

        raise NotImplementedError("Subclasses must implement evaluate()")

**class** Operand(Node):

**def** \_\_init\_\_(*self*, *value*):

*self*.*value* = *value*

**def** evaluate(*self*):

        return *self*.*value*

**class** Operator(Node):

**def** evaluate(*self*):

        raise NotImplementedError("Subclasses must implement evaluate()")

**class** AddOperator(Operator):

**def** evaluate(*self*):

        return *self*.left.evaluate() + *self*.right.evaluate()

**class** SubtractOperator(Operator):

**def** evaluate(*self*):

        return *self*.left.evaluate() - *self*.right.evaluate()

**class** MultiplyOperator(Operator):

**def** evaluate(*self*):

        return *self*.left.evaluate() \* *self*.right.evaluate()

*expression\_tree* = MultiplyOperator()

*expression\_tree*.left = AddOperator()

*expression\_tree*.left.left = Operand(5)

*expression\_tree*.left.right = Operand(4)

*expression\_tree*.right = SubtractOperator()

*expression\_tree*.right.left = Operand(5)

*expression\_tree*.right.right = Operand(2)

*result* = *expression\_tree*.evaluate()

print("Result:", *result*)

