```
"""
Author: Pascal Padilla
Source: correction de l'exercice 2 du sujet 23 des épreuves pratiques NSI 2022

Remarques:
    * classe Pile classique
"""


class Pile:
    """Classe définissant une structure de pile."""
    def __init__(self):
        self.contenu = []

    def est_vide(self):
        """Renvoie le booléen True si la pile est vide, False sinon."""
        return self.contenu == []

    def empiler(self, v):
        """Place l'élément v au sommet de la pile"""
        self.contenu.append(v)

    def depiler(self):
        """
        Retire et renvoie l¿élément placé au sommet de la pile,
        si la pile n¿est pas vide.
        """
        if not self.est_vide():
            return self.contenu.pop()


def eval_expression(tab):
    # création d'une pile vide
    p = Pile()

    # parcours du tableau avec 'element'
    for element in tab:
        # cas d'un nombre -> empiler au sommet de la pile
        # l'élément n'est ni un '+' ET ni un '*'
        if element != '+' and element != '*':
            p.empiler(element)

        # cas d'un opérateur
        else:
            # addition car il y a un '+' dans le résultat
            if element == '+':
                # calcul du résultat
                resultat = p.depiler() + p.depiler()
            else:
                # cas de la multiplication
                # calcul du résultat
                resultat = p.depiler() * p.depiler()

            # empiler le résultat
            p.empiler(resultat)

    # fin BOUCLE
    # tout le tableau est parcouru
    # le résultat est au sommet de la pile
    return p.depiler()


# tests avec des affichages
print(eval_expression([2, 3, '+', 5, '*']))


# tests avec des assertions
assert eval_expression([2, 3, '+', 5, '*']) == 25
```