



Principes de la POO

Programme officiel



Modèle d'architecture
séquentielle (von
Neumann)

constituants d'une machine.
Dérourer l'exécution d'une séquence
d'instructions simples du type langage
machine

RESUMÉ DU COURS

PROGRAMMATION ORIENTÉE

OBJET

Un objet:

- représente une entité physique, un concept...
- est autonome,
- peut interagir avec les autres objets.

La POO consiste à réunir les données caractérisant un objet et leur traitement, dans une unité autonome. On parle **d'encapsulation**.

 À retenir

Une **classe** est une modélisation d'un élément du monde réel.

Une classe possède:

- **des attributs**: il s'agit des caractéristiques spécifiques de l'objet (données),
- **des méthodes**: il s'agit des fonctionnalités que peut réaliser l'objet (traitements et services).

```
class Roche:
    def __init__(self, x: int, y: int):
        self.x = x
        self.y = y
```



```
        return self.x

def get_y(self) -> int:
    return self.y

def get_durete(self) -> int:
    return self.durete

def set_x(self, x: int) -> None:
    self.x = x

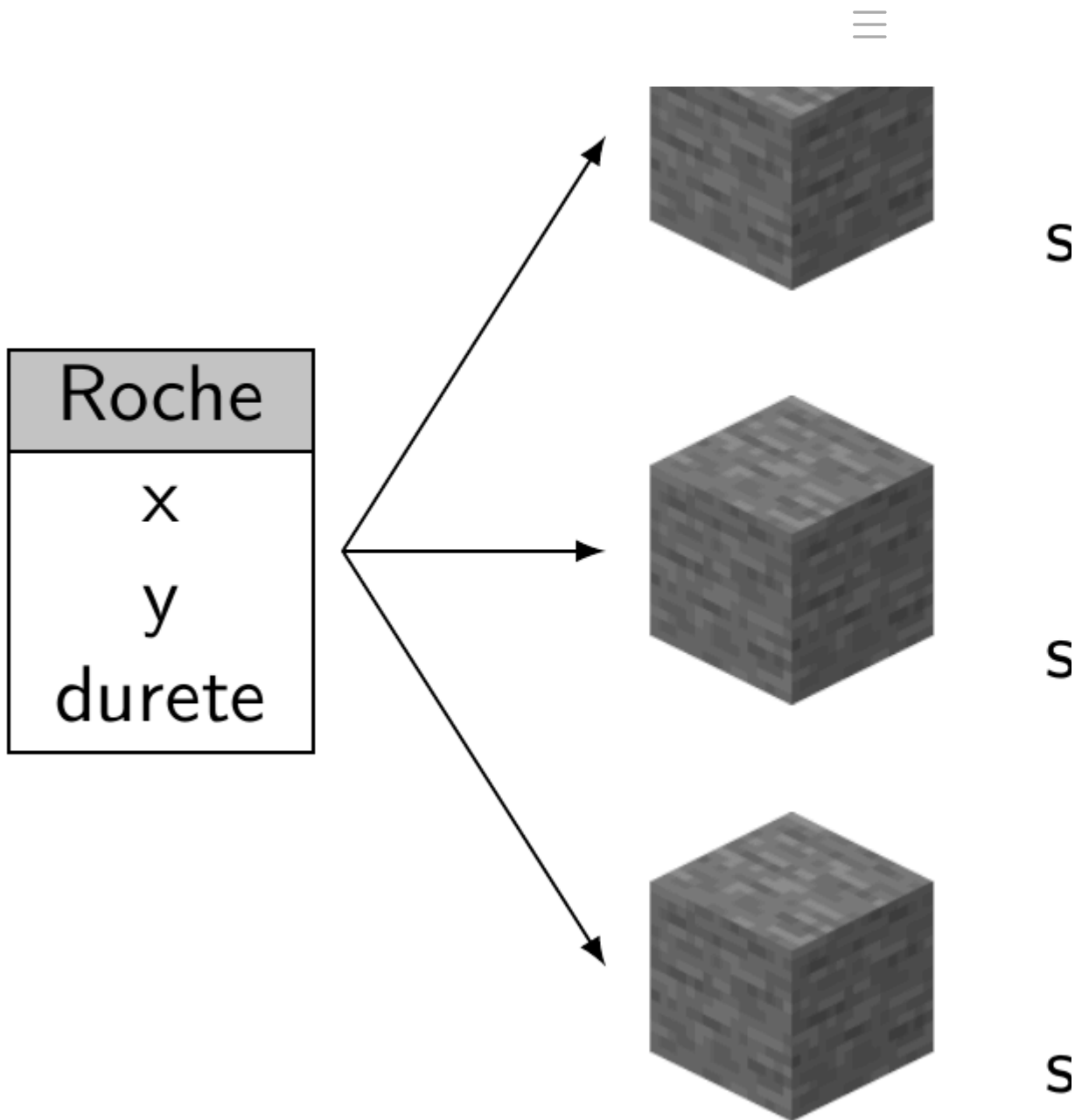
def set_y(self, y: int) -> None:
    self.y = y

def diminuer_durete(self, impact: int) -> None:
    if self.durete > 0:
        self.durete = self.durete - impact
    if self.durete < 0:
        self.durete = 0
```

Certaines méthodes (les **accesseurs** et **mutateurs**) permettent de manipuler les attributs d'un objet.

 À retenir

Une **instance** est une construction concrète, produite à partir de l'objet modèle. L'instance est manipulable dans le programme.



```
bloc_0 = Roche(0, 0)
```

```
bloc_1 = Roche(0, 1)
```



Le résumé en pdf

Il s'agit de cette page exportée en pdf

[Télécharger](#)

Une autre manière d'expliquer

Apprendre la POO avec Graven

[Cliquez-ici](#)

NSI



© Copyright **Selecao**. All Rights Reserved

