

Files et Piles

63-31 - Programmation Collaborative

Les piles et les files

Types abstraits de données, collections, qui ont une interface simple

- ◆ Ajouter un élément à la structure

Pile : push()

File : add()

- ◆ Récupérer un élément de la structure

Pile : pop()

File : remove()

Différence : ordre dans lequel les éléments sont récupérés

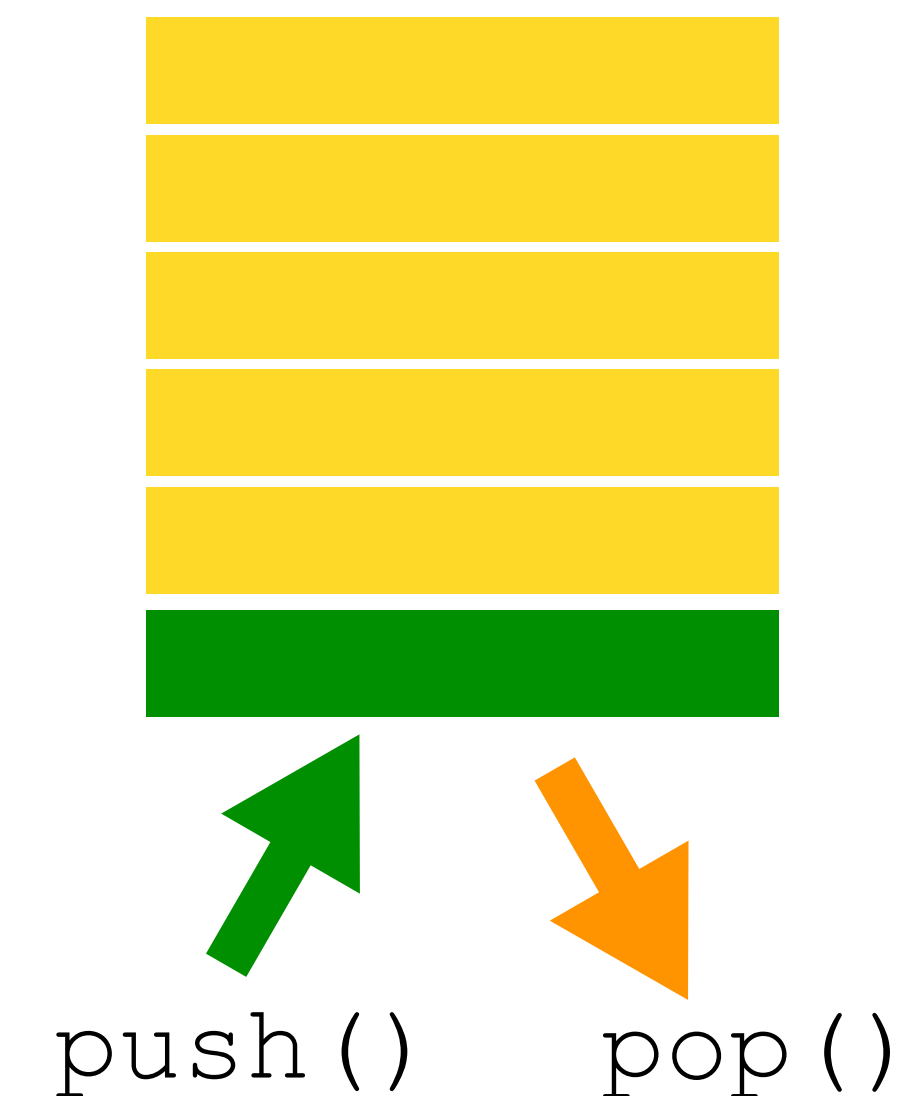
- ◆ File : premier entré, premier sorti

Préserve l'ordre d'arrivée, file d'attente classique

- ◆ Pile : dernier entré, premier sorti

Inverse l'ordre d'arrivée, pile d'assiettes

File (Queue) Pile (Stack)
FIFO - First In First Out LIFO - Last In First Out



Files

- ◆ File à la cafétaria
- ◆ Plus généralement, tous systèmes à attente (paquets IP dans les routeurs, processus pour ordonnancement, ...)

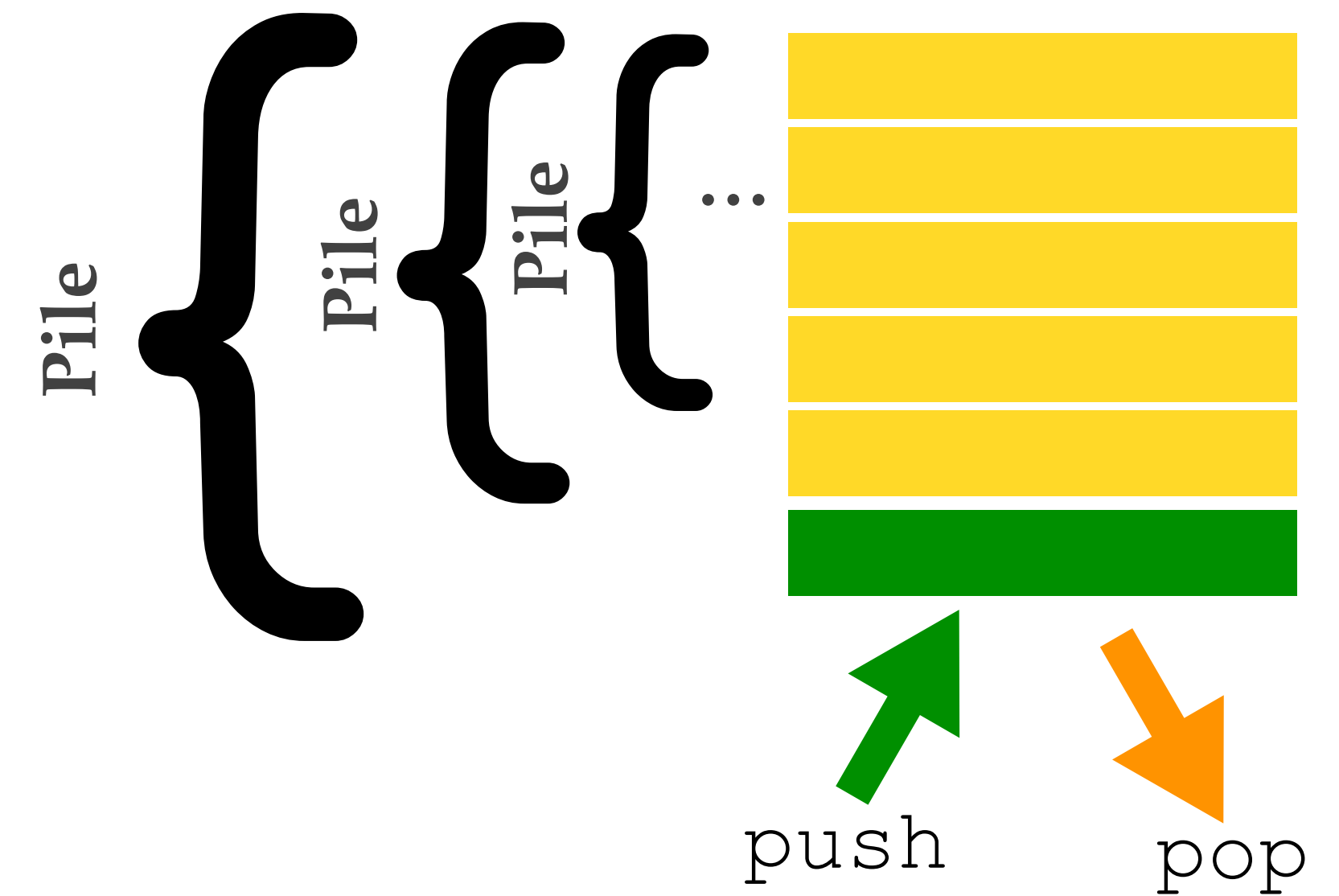
Piles

- ◆ Fonction "Annuler" dans les logiciels
- ◆ Bouton "back" dans les navigateurs
- ◆ Empilement de contexte lors de l'appel de fonctions
- ◆ Jeux : backtracking
- ◆ Graphes : exploration

Quelques détails sur les piles

La pile est, par nature, une structure réursive

♦ Pile = premier element + pile



Un exemple en détails

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

↪ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

↪ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

↪ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push

Evaluation d'expressions arithmétiques postfixe

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

↪ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

↪ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

↪ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



$3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

Evaluation d'expressions arithmétiques postfixe

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$ $\Rightarrow 1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$ $\Rightarrow 3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

$\Rightarrow 3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre \Rightarrow push

Opérateur \Rightarrow pop ; pop ; calcul ; push



3 1 4 + 5 * 4 - 7 / +

push(3)

Evaluation d'expressions arithmétiques postfixe

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

→ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

→ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

→ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



↓
 $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$
push(1)

Evaluation d'expressions arithmétiques postfixe

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$ $\Rightarrow 1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$ $\Rightarrow 3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

$\Rightarrow 3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre \Rightarrow push

Opérateur \Rightarrow pop ; pop ; calcul ; push



↓

$3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

push (4)

Evaluation d'expressions arithmétiques postfixe

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$ $\rightarrow 1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$ $\rightarrow 3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

$\rightarrow 3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre \Rightarrow push

Opérateur \Rightarrow pop ; pop ; calcul ; push



3 1 4 **+** 5 * 4 - 7 / +

pop() \leadsto 4
pop() \leadsto 1
eval(1 + 4) \leadsto 5
push(5)

Exemple d'application

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

↪ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

↪ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

↪ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



$3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

push(5)

Exemple d'application

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

↪ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

↪ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

↪ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



$3\ 1\ 4\ +\ 5\ * \downarrow 4\ -\ 7\ /\ +$

pop() \leadsto 5

pop() \leadsto 5

eval(5 * 5) \leadsto 25

push(25)

Exemple d'application

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

↪ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

↪ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

↪ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



$3\ 1\ 4\ +\ 5\ *\ \downarrow 4\ -\ 7\ /\ +$

push(4)

Exemple d'application

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

→ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

→ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

→ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



$3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

pop() → 4

pop() → 25

eval(25 - 4) → 21

push(21)

Exemple d'application

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

↪ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

↪ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

↪ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



$3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

push(7)

Exemple d'application

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

→ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

→ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

→ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



$3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

`pop() ~ 7`

`pop() ~ 21`

`eval(21 / 7) ~ 3`

`push(3)`

Exemple d'application

Une application : évaluation expressions arithmétiques

◆ $3 + ((1 + 4) * 5 - 4) / 7$

◆ Notation postfixe :

Autre représentation d'une expression arithmétique

$(1 + 4)$

↪ $1\ 4\ +$

$3 + ((1 + 4) * 5 - 4) / 7$

↪ $3\ (((((1\ 4\ +)\ 5\ *)\ 4\ -)\ 7\ /)\ +$

↪ $3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

◆ Permet d'évaluer plus facilement (pour un ordinateur) les expressions en notation postfixe

Nombre ⇒ push

Opérateur ⇒ pop ; pop ; calcul ; push



$3\ 1\ 4\ +\ 5\ *\ 4\ -\ 7\ /\ +$

`pop()` \leadsto 3

`pop()` \leadsto 3

`eval(3 + 3)` \leadsto 6

`push(6)`

Les piles en Java

```
import java.util.Stack;

class TestStack {
    public static void main(String args[]) {
        Stack<Integer> maPile = new Stack<>();
        int x;

        maPile.push(12);
        maPile.push(314);
        maPile.push(212);

        while (! maPile.isEmpty())
        {
            x = maPile.pop();
            System.out.println("--> " + x);
        }
    }
}
```

Conversion d'expression en notation postfixe

Pour chaque élément **x** (nombre, opérateur, parenthèse) de l'expression

si (x entier)

| ⇒ afficher(x)

sinon si (x opérateur)

| tant que priorité(tête(pile)) ≤ priorité (x)

| | y = pop()

| | afficher(y)

| push(x)

sinon si (x == '(')

| push(x)

sinon si (x == ')')

| y = pop()

| tant que tête(pile) ≠ '('

| | afficher(y)

| | y = pop()

Tant que pile non vide

| x = pop()

| afficher(x)

L'algorithme ci-contre permet de convertir une expression "normale" (notation infixe) en notation postfixe

- ◆ Utilisation d'une pile aussi
- ◆ Algorithme pas forcément intuitif
⇒ prendre un exemple (lequel ?)
- ◆ Représentation en pseudo-code

Les files en Java

```
import java.util.queue;

class TestQueue {
    public static void main(String args[]) {
        Queue<Integer> maFile = new LinkedList<>();
        int x;

        maFile.add(12);
        maFile.add(314);
        maFile.add(212);

        while (! maFile.isEmpty())
        {
            x = maFile.remove();
            System.out.println("--> " + x);
        }
    }
}
```

Les files en Java

```
import java.util.queue;
```

```
class TestQueue {  
    public static void main(String args[]) {  
        Queue<Integer> maFile = new LinkedList<>();  
        int x;
```

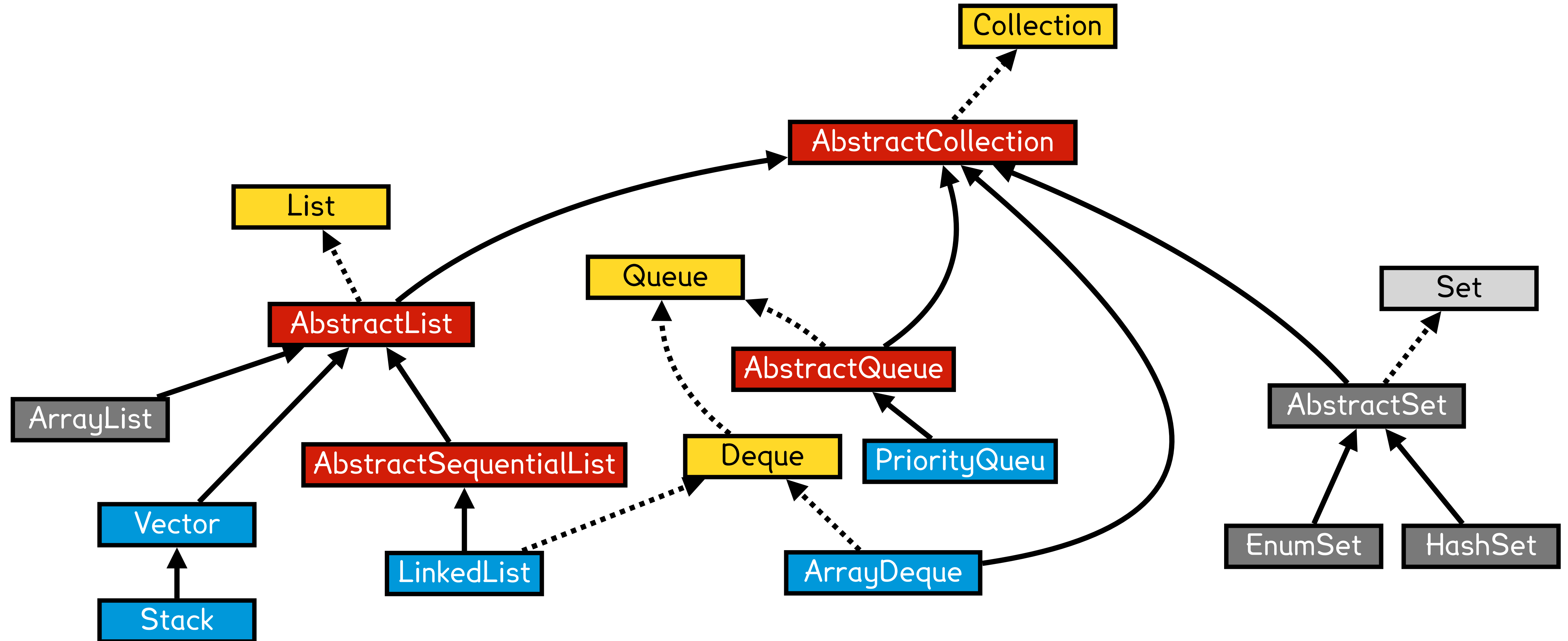
```
        maFile.add(12);  
        maFile.add(314);  
        maFile.add(212);
```

```
        while (! maFile.isEmpty())  
        {  
            x = maFile.remove();  
            System.out.println("--> " + x);  
        }  
    }  
}
```



LinkedList est une classe concrete qui
implémente l'interface Queue

Hiérarchie des collections



Exercice Dirigé

Comment modéliser une exploration de labyrinthe ?

