# Exercices Stream, Optional

On donne une liste de chaînes de caractères. Certaines chaînes représentent des nombres.

public static void main(String[] args) {

List<String> maListe = Arrays.asList(

"1", "deux", "3", "4", "cinq");

}

On demande de faire la somme de ces nombres à l’aide de « streams » et de « lambdas ».

Aide : on peut construire une instance de Stream<Optional<Integer>> via la méthode map et appliquer ensuite la méthode reduce :

public static void main(String[] args) {

List<String> maListe = Arrays.asList(

"1", "deux", "3", "4", "cinq");

Optional<Integer> reduce = maListe.stream()

.map(Main::mapper)

.reduce(Optional.empty(), accu);

}

Les méthodes mapper et accu :

public static void main(String[] args) {

public static Optional<Integer> accu(

Optional<Integer> x, Optional<Integer> y) {

if (x.isPresent()) {

if (y.isPresent())

return Optional.of(x.get() + y.get());

return x;

}

return y;

}

public static Optional<Integer> mapper(String s) {

try {

int parseInt = Integer.parseInt(s);

return Optional.of(parseInt);

}

catch (Exception e) {

return Optional.empty();

}

}