

Exercice 6:

2)

Mon code de tp diffère de celui du TD principalement par la gestion des erreurs dans ajout, qui utilise des exceptions (Exception Planning) au lieu de faire des return false. Je dois donc adapter les tests pour vérifier les exceptions (0 pour une réservation non valide, 1 pour un tableau plein, et 2 pour un doublon.)

- la méthode getReservation return une seule réservation au lieu d'un tableau, ce qui nécessite d'ajuster les tests. (j'ai plus les deux autres méthodes dans mon tp de but1).

3)

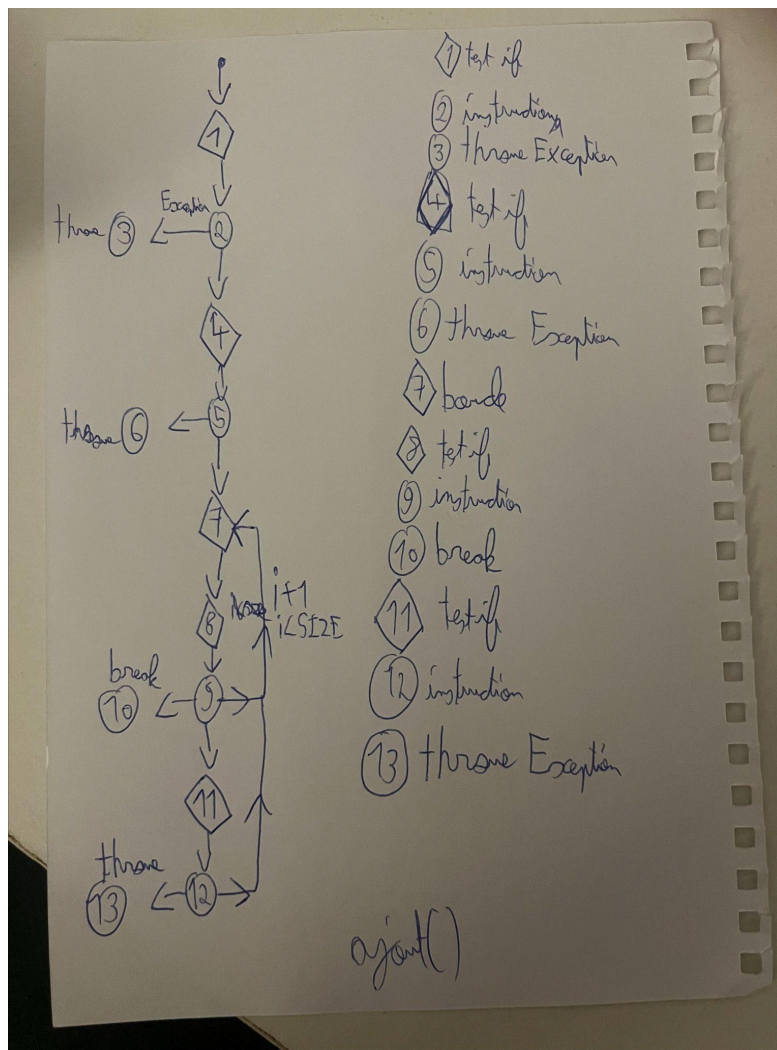
TEST BB AJOUT:

On cherche à concevoir un test pour la méthode ajout en se basant uniquement sur le code source (voir en bas du document) en donné dans mon tp de but1, on utilisera la méthode des chemins indépendants . On va donc analyser le code en identifiant les instructions, les tests, les conditions et les boucles.

entrées:

- chTabReservation un tableau contenant des réservations ayant pour état une plage horaire et une date
- réservation de type Reservation

On réalise le graphe de tâche d' ajout:



C1={1,2,3} C2={1,2,4,5,6} C3={1,2,3,4,5,7,8,9,10} C4={1,2,3,4,5,7,8,9,11,12,13}

C5={1,2,3,4,5,7,8,9,11,12}

On construit le cas test1 dans le tableau 1

Cas Test1: Ajout bb			
test	entrée chTabReservation	entrée reservation	sortie
c1	tableau plein	réservation quelconque	exception
c2	tableau quelconque	reservation non valide	exception
c3	tableau avec au moins une case vide au debut	reservation valide	break
c4	tableau pas plein	reservation déjà existante	exception

c5	tableau pas plein	reservation valide	ajout dans le tableau
----	-------------------	--------------------	-----------------------

Tableau 2: conception test ajout bb:

Cas Test1: tri de données de test bb			
test	entrée chTabReservation	entrée reservation	sortie
c1	[03/02/2025]	05/02/2025	exception
c2	[03/02/2025,null]	34/02/2025	exception
c3	[null,03/02/2025]	03/03/2025	break
c4	[03/02/2025,...] size=8	03/03/2025	exception
c5	[03/02/2025,...] size=8	05/02/2025	ajout dans le tableau

```

public void ajout(Reservation reservation) throws
ExceptionPlanning {
    if (chTabReservations[TAILLE - 1] != null) {
        System.out.println("tableau plein");
        throw new ExceptionPlanning(1);
    }
    if (!reservation.estValide()) {
        System.out.println("reservation non valide");
        throw new ExceptionPlanning(0);
    }
    for (int i = 0; i < chTabReservations.length; i++) {
        if (chTabReservations[i] == null) {
            chTabReservations[i] = reservation;
            break;
        }
        if (chTabReservations[i].compareTo(reservation) == 0) {
            System.out.println("reservations non compatibles");
            throw new ExceptionPlanning(2);
        }
    }
}
}

```

TEST BB getReservation:

On cherche à concevoir un test sur la méthode getReservation en ne s'appuyant uniquement sur le code source (voir bas du document) pour cela on utilise la méthode des chemins indépendants. On identifie les boucles, instructions, conditions et tests.

Entrées:

- date de type Date
- chTabReservation tableau contenant des réservations ayant pour état une plage horaire et une date

sortie:

Une réservation de type Reservation

on réalise le graphe de tâche de getReservation

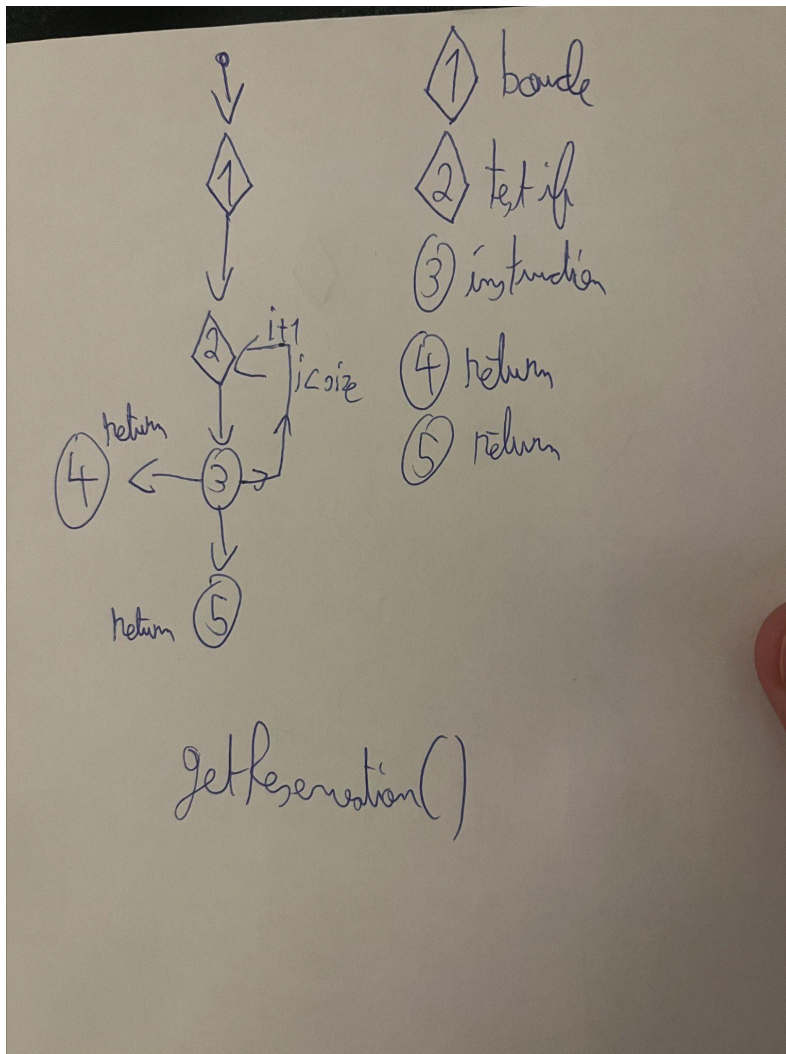


Figure 2: graphe de tâche de getReservation:

$c1=\{1,2,3,4\}$ $c2=\{1,2,3,5\}$ $c3=\{1,2,3,2,3,4\}$ $c4=\{1,2,3,2,3,5\}$
On a identifié 4 chemins.

On construit le "cas test2" dans le tableau 3:

Cas test2: getReservation bb			
test	entrée date	entrée chTabReservation	sortie Reservation
c1	date valide	tableau non null	réservation à la date entrée
c2	date quelconque	size=0	null

c3	date valide	tab avec size>1	réservation à la date entrée
c4	date valide	tab avec size>1	null car réservation non trouvée

Tableau 3: conception test getReservation bb:

Cas test2bis: getReservation données de test bb			
test	entrée date	entrée chTabReservation	sortie Reservation
c1	13/02/2025	[13/02/2025]	13/02/2025
c2	17/03/2005	[]	null
c3	17/03/2005	[17/03/2025,12/03/2025]	17/03/2005
c4	date valide	tab avec size>1	null

```

public Reservation getReservation(Date date){
    for (int i = 0; i < chTabReservations.length; i++){
        if (chTabReservations[i] != null &&
chTabReservations[i].chDate.equals(date)){
            System.out.println("trouvé");
            return chTabReservations[i];

        }
    }
    return null;
}

```