# Etude de cas : Programmation système

Table des matières

[Etude de cas : Programmation système 1](#_Toc154060613)

[Question 1 : 2](#_Toc154060614)

[A 2](#_Toc154060615)

[B 2](#_Toc154060616)

[C 2](#_Toc154060617)

[Question 2 : 2](#_Toc154060618)

[A 2](#_Toc154060619)

[B 2](#_Toc154060620)

[C 2](#_Toc154060621)

[Question 3 : 2](#_Toc154060622)

[A 2](#_Toc154060623)

[B 2](#_Toc154060624)

[Question 4 : 3](#_Toc154060625)

[Question 5 : 3](#_Toc154060626)

[1 3](#_Toc154060627)

[2 3](#_Toc154060628)

[3 3](#_Toc154060629)

[4 3](#_Toc154060630)

[5 3](#_Toc154060631)

## Question 1 :

### A

Le bloc 1 créer un DP mutex qui permet de suivre l’état des places de parking (libre ou occupé)

### B

Le bloc 2 vérifie si le parking est complet.

L’attribut \_nb\_cars a une syntaxe plutôt propre au get/set on pourrait donc lui donner une syntaxe de variable en l’appelant nb\_cars.

### C

Le bloc 3 bloque tant qu’un thread ne s’est pas libéré.

## Question 2 :

### A

Oui il l’impose par le biais du protocole TCP.

### B

Oui c’est sa fonction principale

### C

Non car elle utilise forcément le port 8888 donc on ne pourra pas s’en servir plusieurs fois sur plusieurs consoles.

## Question 3 :

### A

Un thread.sleep est utilisé, comme on est dans un cas d’attente un await aurait était plus adapté.

### B

Les 2 premières variables : nbMax\_parking\_spot et\_nbMax\_car\_parking\_spot vont 2 paires la 2nd est uniquement utilisé pour définir la taille max de la 1er comme ça demande pas d’entré de l’utilisateur on aurait pu imaginer directement déclarer nbMax\_parking\_spot = 10.

nb\_car\_in\_park : est une variable de suivi utilisé plusieurs fois dans le code, elle me semble donc être déclaré de manière pertinente.

t\_parking\_spot : comme la première variable elle aurait pu directement être instancié.

## Question 4 :

La classe est fonctionnel on pourrait cependant améliorer le formatage des donnés dans le fichier csv.

## Question 5 :

### 1

Il n’y a visiblement pas de convention de nommage sur le nommage de variable ce qui rend la relecture du code complexe, tout comme l’absence de commentaire.

### 2

La méthode : NbVacant\_parking\_spot n’est utilisé que 2 fois est n’est qu’un Get ce qui n’est pas une bonne pratique de POO.

### 3

Pour l’exécution du programme, une partie est dans le socket et une partie est dans le program.cs ce qui n’est pas très logique.

### 4

Release\_parking\_spot n’est utilisé qu’une seule fois dans Exit\_gates on pourrait donc supprimer la méthode est mettre le code en dur.

### 5