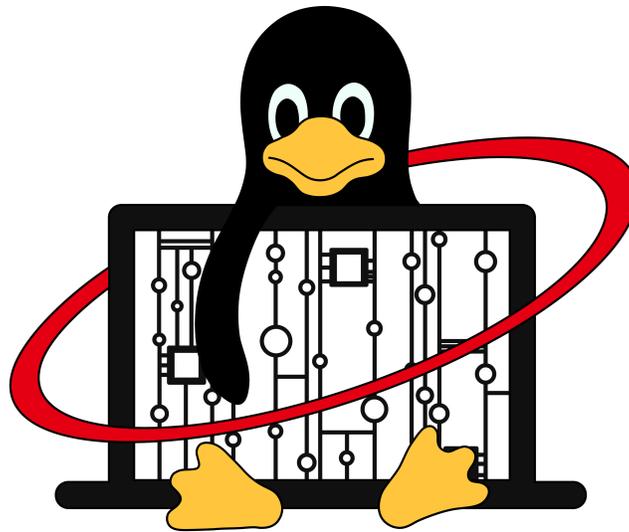


La tournée de Taylor Swift

Paul Slisse

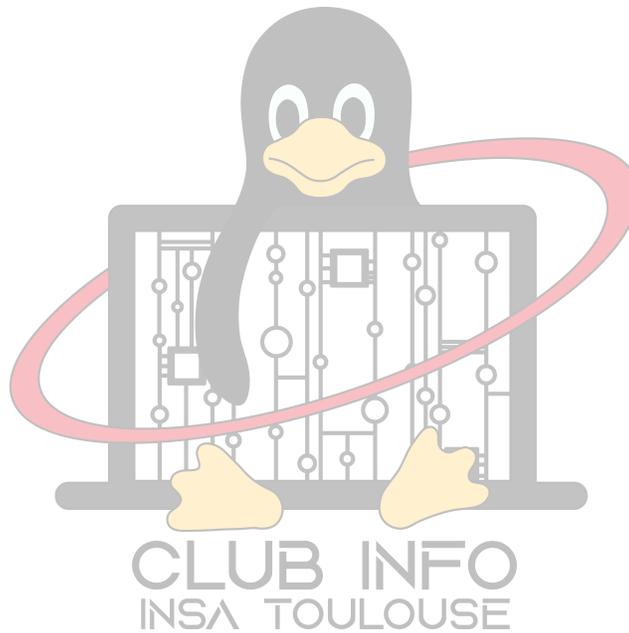
28 avril 2024



CLUB INFO
INSA TOULOUSE

Table des matières

1	Énoncé	3
2	Aide	5
3	Correction	6
3.1	File_Gen.ads	6
3.2	File_Gen.adb	6
3.3	Main.adb	6



1 Énoncé

Cette charmante Taylor Swift n'a pas encore finie sa tournée qu'elle veut déjà préparer la prochaine. Pour cela, elle veut un nouveau site web rien que pour cette tournée. Vous avez postulé pour cette offre inratable et vous avez été enrôlé dans l'équipe de développement.

Vous avez été assigné à la tâche de gestion des connexions au serveur. Pour éviter que le serveur ne plante lors de l'ouverture de la vente, vous avez décidé de créer un système de file d'attente de connexion pour gérer les accès et ne pas surcharger le serveur.

Question 1 Définition d'un type *Utilisateur*

Tout d'abord, créez un fichier principal dans lequel vous gérerez par la suite la file d'attente. Premièrement, on va définir un type *Utilisateur* qui décrira les personnes souhaitant se connecter. On utilisera :

IP : Integer
Login : String(1..6)
Code : Integer

Le code correspond à un code de connexion défini par l'utilisateur pour se connecter et le login à une chaîne de 6 caractères elle aussi défini par l'utilisateur (des fonctionnalités mis en place par votre voisin de bureau).

Question 2 Package de Files générique

Vous êtes tellement sympa que vous vous dites que vous allez faire un package générique de file d'attente au cas où vous ou un de vos collègues en aurait encore besoin par l'avenir.

Créez un package *File_Gen* dans lequel vous implémenterez les différentes opérations dont vous avez besoin. On veut pouvoir créer des Files de taille maximum différentes pour en avoir une pour les gens qui sont actuellement connectés et ceux qui le souhaitent (=attente). Ainsi, pour instancier le package de File, on veut pouvoir donner un type de données ainsi qu'un nombre qui donnera la taille max (ce seront les deux arguments de généricité). Créez le *.ads* en définissant le type File à l'aide de 2 pointeurs (un sur le début et un sur la fin) et la hauteur de la file. On implémentera les opérations *Init*, *Est_Vide*, *Enfiler*, *Lire* et *Defiler*. On déclarera également deux exceptions : *File_Vide* et *File_Pleine*.

Aide : Si vous avez du mal pour cette question, notamment pour bien définir le type File, je vous invite à vous référer à la section *Aide* pour avoir la correction uniquement de l'implémentation du type File.

Question 3 Implémentations des opérations

Le plus gros du boulot, implémenter les différentes opérations : *Init* (Initialisation d'une file vide), *Est_Vide* (Test si une file est vide ou non), *Enfiler* (Ajoute un élément dans la file, attention à la taille max, \Rightarrow *Exception*), *Lire* (qui renvoie la valeur et tte de la liste, \Rightarrow *Exception*) et *Defiler* (Retire un élément de la file, \Rightarrow *Exception*). On oublie pas de s'assurer que la file n'est pas vide avant de retirer un élément.

Question 4 Création des 2 files

Dans votre fichier principal, instanciez 2 files d'utilisateurs, une d'attente avec une taille max très grande et une de connexion plus petite (ici on prendra 5 pour tester plus facilement).

Question 5 Affichage d'une file

Modifiez le `.ads` et `.adb` pour ajouter une procédure d'affichage d'une file pour que vous puissiez en suivre l'avancement.

Aide : On peut utiliser l'attribut `'Image` sur n'importe quel type :)

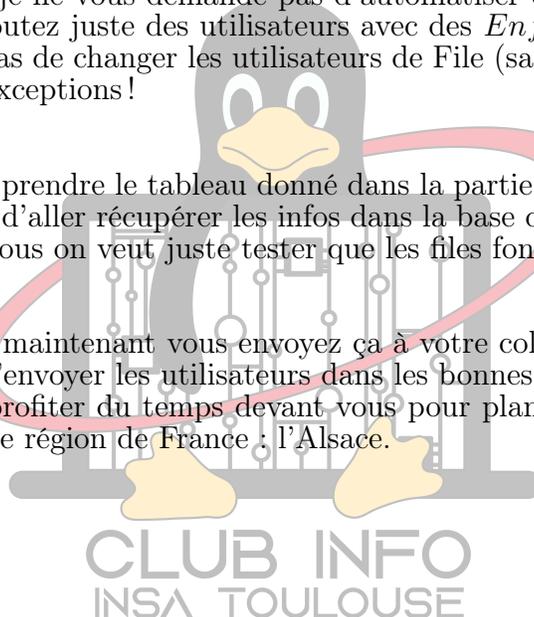
Question 6 Simulation

Dans le programme principal, on va simuler l'arrivée et le départ de plusieurs utilisateurs sur la plateforme de vente. On veut pouvoir vérifier que le système fonctionne bien, ainsi si la file de connexion est remplie, on va ajouter l'utilisateur à la file d'attente jusqu'à ce qu'une place se libère. On utilisera la procédure d'affichage de la question précédente pour vérifier le remplissage.

Ne vous inquiétez pas, je ne vous demande pas d'automatiser ça, c'est un de vos collègues qui le fera :), ajoutez juste des utilisateurs avec des `Enfiler` et `Defiler`. On ne s'occupera également pas de changer les utilisateurs de File (sauf si vous êtes vraiment déter). Attention aux exceptions!

Aide : Vous pouvez prendre le tableau donné dans la partie *Aide*. Pareil c'est votre collègue qui s'occupera d'aller récupérer les infos dans la base de données pour ajouter ensuite dans les files. Nous on veut juste tester que les files fonctionnent.

Youpiiiii, ça marche maintenant vous envoyez ça à votre collègue en charge d'automatiser le système et d'envoyer les utilisateurs dans les bonnes files. Vous décidez donc de profiter du temps devant vous pour planifier vos prochaines vacances dans la plus belle région de France : l'Alsace.



2 Aide

Question 2 : Déclaration du type File

```
1 generic
2   type Element_Type is private;
3   Max_Taille : Positive;
4
5 package File_Gen is
6   type File is limited private;
7   .
8   .
9   .
10 private
11   type Cellule;
12   type Pointeur is access Cellule;
13   type Cellule is record
14     Info : Element_Type;
15     Suivant : Pointeur;
16   end record;
17   type File is record
18     Debut : Pointeur;
19     Fin : Pointeur;
20     Hauteur : Natural;
21   end record;
22 end File_Gen;
```

Question 6 : Tableau d'utilisateurs

```
1 Utilisateurs : Utilisateur_Array(1..10) := ((IP => 192168001, Login => "user01", Code => 123456),
2 (IP => 192168002, Login => "user02", Code => 654321),
3 (IP => 192168003, Login => "user03", Code => 987654),
4 (IP => 192168004, Login => "user04", Code => 246810),
5 (IP => 192168005, Login => "user05", Code => 135792),
6 (IP => 192168006, Login => "user06", Code => 112233),
7 (IP => 192168007, Login => "user07", Code => 445566),
8 (IP => 192168008, Login => "user08", Code => 778899),
9 (IP => 192168009, Login => "user09", Code => 987123),
10 (IP => 192168010, Login => "user10", Code => 456789));
```

3 Correction

3.1 File_Gen.ads

1 Bla

3.2 File_Gen.adb

1 Bla

3.3 Main.adb

1 Bla

