

[C++] PRALG : Rapport sur le TP Arbres, À l'attention de monsieur Pascal Monasse

Voici les 6 types d'erreurs que j'ai identifiées. Pour chacune d'entre elles nous allons voir pour lesquelles on peut facilement signaler l'erreur par valeur de retour ou ajouter un statut d'erreur. Pour lesquelles on peut signaler l'erreur par exception. Justifier notre choix de signalement d'erreur.

Risque que le joueur tente de mettre une valeur non entière dans un nœud

Signaler l'erreur par valeur de retour ou ajouter un statut d'erreur ?

Pas de valeur de retour, juste un message renvoyé grâce à `cerr` pour signaler l'erreur.

Choix et Justification du signalement ?

Try catch dans le main. Nous avons décidé que cette gestion d'erreur devait être faite par l'utilisateur pour que l'utilisateur du programme, d'où le choix de rédaction dans le main.

Risque d'Out of range dans le vecteur (Setson)

Signaler l'erreur par valeur de retour ou ajouter un statut d'erreur ?

Par valeur de retour : `exit(errno);`. En effet, ce type d'erreur est directement géré par le vecteur qui alloue de l'espace dynamiquement.

Choix et Justification du signalement ?

Try catch directement dans la méthode SetSon grâce au `at` qui lève une exception en cas de soucis. Donc rédigée par le développeur. Le catch est muni de la mention Out of range qui répond aux appels de type out of range.

Risque d'Underflow : Utilisation de `RemoveLastSon` sans qu'il y ait d'autres sons à enlever

Signaler l'erreur par valeur de retour ou ajouter un statut d'erreur ?

Par statut d'erreur : `if (sons.size()==0) throw underflow_error.`

Choix et Justification du signalement ?

Le risque d'Underflow est directement géré par le vecteur utilisation d'un try catch

Risque que `Getson` retourne un pointeur nul

Signaler l'erreur par valeur de retour ou ajouter un statut d'erreur ?

Choix et Justification du signalement ?

Nous avons simplement utilisé un `if (pos>=int(sons.size())) throw out_of_range.` En effet, le but étant que le développeur sache d'où provient l'erreur, nous mettons un message spécifique pour qu'il comprenne.

Get data : entier non initialisé

Choix et Justification du signalement ?

```
if (data>INT8_MAX) throw out_of_range
```

Lorsque le d n'est pas initialisé, il est pris de manière aléatoire plus grand que INT8_MAX.