

TPO3 BIS – LES ALTERNATIVES SUITES.

OBJECTIFS

Utiliser les structures alternatives permettant :

- De contrôler le déroulement d'un programme en fonction de choix fait par l'utilisateur, ou d'autres critères.
- De contrôler la saisie de l'utilisateur : vérification.

APPRENDRE À LIRE UN ÉNONCÉ – PARTIE 1

Remarque 1 :

Dans les sujets qui suivent parfois celui-ci ne donnera aucune indication sur le type de donnée à choisir. Lorsque le sujet ne le précise pas et dans ce cas, ce choix est laissé à la réflexion du programmeur. Il faudra choisir ce qui est le plus adapté. Cela demande réflexion. Le type « int » n'est pas l'unique choix.

Remarque 2 : Dans certains énoncés, vous aurez des indications importantes. Par exemple, un sujet pourrait spécifier que des notes d'élèves sont comprises entre 0 et 20. Avec cette précision, vous savez ce qu'il faut vérifier dans vos tests.

Cette précision ne sera pas systématique dans tous les énoncés. Bien souvent, c'est au programmeur à réfléchir au contexte du sujet et à voir dans quelle condition la saisie est valide.

Même si aucune vérification n'est demandée clairement dans le sujet il faudra ajouter ses contrôles de saisies. Cela fait partie des compétences attendues d'un programmeur.

Dans les exercices qui suivent, il faut tenir compte des 2 remarques précédentes.

Rappelez-vous également de l'objectif travaillé : les alternatives sont utilisées pour contrôler le déroulement d'un programme, ou vérifier les saisies utilisateurs ou bien les 2 en même temps.

Les algorithmes auront la plupart du temps la structure suivante :

```
SAISIR les Variables
Si VÉRIFICATION en échec alors
    Afficher ERREUR + EXPLICATION
Sinon
    'tout est OK
    Exécution du programme en utilisant éventuellement des alternatives
Finsi
```

EXERCICES

Nous faisons évoluer notre gestion des projets sous NetBeans. Plutôt que d'avoir un fichier .java par exercice nous allons tout regrouper dans un seul fichier. Chaque exercice sera une fonction comme suit :

```
public static void exercice1() {
    //code exercice 1 here
}
```

Le programme principal appellera l'exercice voulue de la façon suivante :

```
public static void main(String args[]) {
    exercice1();
}
```

EXERCICE 1 :

Écrire un programme qui demande l'âge d'une personne, puis affiche un des 2 messages : « vous êtes majeur » ou « vous êtes mineur ».

Pour vous aider sur ce premier exercice : on peut imaginer qu'un âge valide c'est >0 et <120.

EXERCICE 2A :

On veut écrire un programme de calcul d'heure. L'utilisateur tape l'heure et les minutes, et le programme affiche l'heure qu'il sera une minute plus tard. Par exemple, si l'utilisateur tape 21 puis 32, l'algorithme doit répondre :

"Dans une minute, il sera 21 heure(s) 33"

EXERCICE 2B :

Même exercice que précédemment, mais maintenant on ajoute maintenant les secondes saisies par l'utilisateur. Le calcul de l'heure se fait maintenant une seconde plus tard.

EXERCICE 3A:

Écrire un algorithme permettant de saisir une date avec 3 nombres représentant le jour, le mois et l'année. Le programme vérifie si celle-ci est valide

Vérification simple : $0 < J < 32$ et $0 < M < 13$, et $A > 0$

EXERCICE 3B : LA DATE, LE RETOUR

Même exercice que précédemment avec une vérification plus complexe : en fonction du mois celui-ci comporte 30,31 ou 28 ou 29 jours

Cet exercice est certes d'un manque d'originalité affligeant, mais après tout, en algorithmique comme ailleurs, il faut connaître ses classiques ! Et quand on a fait cela une fois dans sa vie, on apprécie pleinement l'existence d'un type « Date » dans certains langages.

Il n'est sans doute pas inutile de rappeler rapidement que le mois de février compte 28 jours, sauf si l'année est bissextile, auquel cas il en compte 29. L'année est bissextile si elle est divisible par quatre. Toutefois, les années divisibles par 100 ne sont pas bissextiles, mais les années divisibles par 400 le sont.

EXERCICE 4 : LE CHOIX MULTIPLE

On commence à avoir pas mal d'exercice à gérer. On va simplifier l'utilisation de notre application en proposant à l'utilisateur un menu permettant de faire le choix de l'exercice à exécuter.

Il prend la forme suivante :

1. Exercice 1
2. Exercice 2
3. Exercice 3
4. Exercice 4

Quand l'utilisateur choisit 4, exécute l'exercice 4.

EXERCICE 5

L'utilisateur saisit 2 nombres entiers au clavier. Le programme affiche ensuite l'un des 2 messages suivants : "le produit des 2 nombres est positif ou nul" ou "le produit des 2 nombres est négatif".

Attention : le produit des 2 nombres ne doit pas être calculé.

Exemple :

A ? 10

B ? 20

Le produit est positif.

A ? 10

B ? -22

Le produit est négatif.

EXERCICE 6

On souhaite écrire une application permettant d'aider un utilisateur à calculer le cout de sa cotisation à un club de basket.

La cotisation de base est donnée par le tableau suivant :

Catégorie	Année de naissance	Prix
U7 BABY BASKET	2016, 2017 et 2018	93 euros
U9	2014 et 2015	112 euros
U11	2012 et 2013	120 euros
U13	2010 et 2011	132 euros
U15	2008 et 2009	132 euros
U17 Masculins	2006 et 2007	150 euros
U18 Filles	2005, 2006 et 2007	150 euros
U20 Masculins	2003-2004-2005	150 euros
SÉNIORS	2002 et avant	155 euros
LOISIRS	2003 et avant	96 euros

Utiliser un avantage carte :

- À ce prix peut être déduite la somme de 20€ si l'adhérent dispose de la carte top départ (pour les collégiens). Même s'il possède cette carte il n'a pas d'obligation d'utiliser cette somme pour la cotisation. (<https://www.ladrome.fr/actualites/rentree-2021-top-depart/>)
- Si vous avez la carte pass'Region (pour les lycéens) dans ce cas il est possible de déduire 30€ sur la cotisation. (<https://fr.calameo.com/read/00011978142e534f25425>)

Enfin vous pouvez avoir une réduction de 50€ proposé par la région sous certaines conditions énumérées ici : <https://crosauvergnerrhonealpes.fr/le-passsport-auvergne-rhone-alpes/>. Cette réduction peut se cumuler avec les avantages carte citée précédemment.

D'autre part, le club demande une participation de 15€ supplémentaire si l'adhérent n'habite pas la commune du club.