

Terminale STMG > Sciences de gestion et numérique > Programmation structurée :
les séquences



Programmation structurée : les séquences

 [Fiche de cours](#)

 Vidéos

 Quiz

 Profs en ligne

 [Télécharger
le pdf](#)

Objectif :

Pourquoi avoir une approche méthodologique du développement des programmes ?

1. Présentation : définition des éléments de base de la programmation

Rappel : le **cycle de vie des logiciels** correspond à une **démarche méthodologique** de résolution de problèmes à partir de l'abstraction des informations du système vers la réalisation d'applications informatiques répondant aux besoins des utilisateurs.

Les éléments de la « **programmation structurée** » ont été proposés aux développeurs des applications informatiques de façon à pouvoir intégrer une **approche méthodologique à la conception des programmes et logiciels**.

L'objectif poursuivi est de **faciliter le développement et la maintenance des logiciels**.

2. Les bases de la programmation structurée

La programmation structurée repose sur le concept que tout programme peut être développé uniquement à partir de trois constructeurs :

- **la séquence** (enchaînement en séquence de traitement),
- **l'alternative** (instruction conditionnelle : traitement permettant d'effectuer un choix),
- **la répétition** (itération ou boucle : traitement pouvant être reproduit plusieurs fois).

application informatique

Sachant qu'il existe de nombreuses méthodes de développement d'une application informatique, nous allons présenter celle qui correspond le mieux à l'approche de la résolution de problèmes simples issus de la réalité du système d'information des organisations.

a. La méthode

La méthode est la **décomposition fonctionnelle** qui correspond à décomposer un problème en ses différentes fonctionnalités de telle façon à simplifier sa résolution.

En décrivant les **structures de données** (fonction « stocker » du système d'information) et en appliquant un **traitement modulaire** (fonction « traiter » du système d'information).

b. La démarche

La démarche repose sur les étapes du cycle de vie des logiciels :

- La **spécification du problème**, qui correspond à une définition, précise les objectifs à atteindre (soit l'énoncé du problème dans une vision scolaire).
- La **conception abstraite** est une étape de réflexion qui correspond à une étude permettant de définir les composants du problème précédemment spécifié soit :
 - les données du problème (informations dont on a besoin pour le résoudre),
 - les résultats du problème (informations qui seront restituées à l'issue des traitements à l'utilisateur),
 - les traitements selon une démarche ascendante (des données vers les résultats) ou descendante (des résultats décomposés fonctionnellement pour en extraire les données).
- La **conception concrète** qui correspond en une approche algorithmique qui a pour rôle de définir :
 - un lexique présentant les structures de données,
 - l'ordonnancement des traitements (ordre de traitement des instructions du programme).
- La **codification dans un langage de programmation** intégrant :
 - la définition de la maquette de l'application,
 - la traduction dans un langage de programmation,
 - le contrôle de la syntaxe.
- Les **tests** qui correspondent :
 - au choix d'un échantillon caractéristique de données représentant la réalité du système d'information,
 - à l'exécution du programme à partir de l'échantillon précédent des données qui va permettre de valider l'application

4. Illustration d'un problème de séquence par le déroulement de la démarche

a. Spécification (énoncé du problème)

On calcule le montant toutes taxes d'une facture à partir du montant net commercial et du taux de TVA de 20%.

b. Conception abstraite (réflexion sur les constituants du problème)

A partir de l'énoncé, on doit définir les **flux entrants** (les données du problème), les **flux sortants** (les résultats du problème), et le moyen de passer des uns aux autres.

Soit :

On calcule le **montant toutes taxes d'une facture** à partir du **montant net commercial** et du **taux de TVA de 20%**.

Ce qui donne les traitements suivants :

le **montant toutes taxes d'une facture** = **montant net commercial** + **montant de la taxe**

avec **montant de la taxe** = **montant net commercial** x **taux de TVA de 20%**.

c. Conception concrète (approche algorithmique du problème)

A partir de l'étape précédente on va devoir définir le lexique associé à l'ordonnancement des instructions du traitement.

<u>Lexique</u>		
<u>Constante</u>		
TxTVA = 20%	le taux de TVA	
<u>Variable</u>		
MONTTC	(réel)	le montant toutes taxes de la facture
NETCOM	(réel)	le montant net commercial
<u>Début</u>		
Lire(NETCOM)		
MONTTC ← NETCOM * (1 + TxTVA)		
Ecrire(MONTTC)		
<u>Fin</u>		

On se place sous l'angle de vue de l'ordinateur qui exécute les programmes (tout en restant indépendant d'un quelconque langage de programmation) :

- la procédure **Lire()** correspond à la primitive permettant aux programmes d'**acquérir les informations** qui seront fournies par l'utilisateur,
- la procédure **Ecrire()** correspond à la primitive permettant aux programmes de **restituer les résultats** à l'utilisateur.

d. Codification (programmation en Visual Basic pour ACCESS)

On peut choisir un (ou des) langage(s) de programmation cible(s) dans le(s)quel(s) on va traduire les éléments précédemment conçus (à la phase de conception).

Exercice 2 :
On calcule le montant toutes taxes d'une facture à partir
du montant net commercial et du taux de TVA de 20%

le montant net commercial

Calcul

le montant toutes taxes

Zone texte : TextNETCOM

Bouton : BtCALCUL

Zone texte : TextMONTTC

La maquette correspond à un ou des formulaires qui seront constitués à partir de contrôles graphiques (composants graphiques) dont certains renferment des procédures événementielles qui seront activées par l'utilisateur (habituellement un clic de souris).

• Code Visual Basic pour ACCESS

```
'clause de demande de déclaration des variables
Option Explicit
'déclaration de la constante taux de taxe
Const TxTAXE = 0.2

Private Sub BtCalcul_Enter()
'déclaration des variables (locales)
Dim NETCOM As Double
Dim MONTTC As Double

'recupération des données
NETCOM = CDb1(TextNETCOM.Value) 'lire(MONTHT)
'traitement
MONTTC = NETCOM * (1 + TxTAXE)
'restitution du résultat
TextMONTTC.Value = CStr(MONTTC) 'écrire(MONTTC)
End Sub
```

e. Tests (jeux d'essai de validation de l'application)

Une application n'est définitivement valide qu'après avoir subi une phase de tests par l'intermédiaire d'un choix judicieux de jeux d'essai (échantillon de données correspondant à la réalité du problème à traiter).

5. Vocabulaire de la programmation structurée

• **La séquence** est une instruction simple d'un programme comme :

- un calcul : $MONTTC \leftarrow NETCOM * (1 + TxTVA)$,
- l'acquisition d'une donnée : Lire(NETCOM),
- la restitution d'un résultat : Ecrire(MONTTC) ,
- ou l'utilisation d'une fonction intrinsèque/prédéfinie : $Y \leftarrow \text{racine_carrée}(X)$.

• **Un type informatique** permet de définir une variable dans son domaine de valeur. Les types informatiques standards sont : réel, entier, chaîne de caractères, caractère, booléen (vrai ou faux), mais aussi structures ou encore tableau...

• **Une variable** est une information (donnée, résultat, variable de travail) dont la valeur change en fonction de l'exécution du programme. Une variable se doit d'être typée en fonction du type informatique proposé par les langages de programmation comme : MONTTC (réel), NOM(chaine), ERREUR (booléen),.

La définition de constantes permet de faciliter la maintenance.

- **L'ordonnancement** des instructions d'un programme consiste en la définition logique de l'ordre d'exécution des différentes instructions comme : acquérir les données du problème puis calculer les résultats et enfin restituer les résultats.
- **La maquette** d'une application correspond à l'ensemble des formulaires qui seront nécessaires pour l'application ; chaque formulaire est construit à partir de contrôles graphiques qui pourront renfermer des procédures événementielles.
- **Une procédure événementielle** correspond au traitement (ensemble des instructions d'une partie du programme) qui seront activées habituellement par l'utilisateur grâce à un événement (le plus souvent un clic de souris).
- **Le commentaire** est un élément facultatif d'un programme qui permet de le commenter de façon à le rendre plus lisible et donc à en faciliter la maintenance.

L'essentiel

La programmation structurée est une méthode permettant le développement cohérent d'applications simples et reposant sur trois constructeurs : la séquence, l'alternative, la répétitive.

Une application informatique est décomposée en deux parties :

- la **maquette** (ensemble des formulaires) qui représente l'interface homme machine mise à disposition de l'utilisateur,
- les **traitements programmés** permettant la gestion des flux de données et la restitution des résultats.

Les **séquences** regroupent les instructions :

- d'**acquisition** des données d'un problème (primitive algorithmique lire(variable)),
- de **restitution** des résultats d'un problème (primitive algorithmique écrire (variable)),
- de **traitement** des informations d'un problème,
- d'**utilisation des fonctions** (ou procédures) intrinsèques ou prédéfinies des langages de programmation.

Comment as-tu trouvé ce cours ?

Évalue ce cours !



Sois le premier à évaluer ce cours !

Maxicours

Exerce toi en t'abonnant



Fiches de cours les plus recherchées

[Les réseaux informatiques](#)

[Coopération entre les applications
\(Middleware\).](#)

[Gestion du projet c
d'informati](#)

[Le système d'information](#)

[Fonctionnalités et cycle de vie de
l'application](#)

[Définition des objet
de donnée](#)



DES PROFS EN LIGNE



DES RESSOURCES RICHES



DES OUTILS LUDIQUES

- Coach virtuel
- Quiz interactifs
- Planning de révision



DES TABLEAUX DE BORD

- Suivi de la progression
- Score d'assiduité
- Une interface Parents

S'abonner

Utilisez l'app pour réviser partout



Cycles

[Primaire](#) [Collège](#) [Lycée](#)

Nos cours par classe

[CP](#) [CE1](#) [CE2](#) [CM1](#) [CM2](#) [6eme](#) [5eme](#) [4eme](#) [3eme](#) [Seconde](#)
[Premiere](#) [Terminale S](#) [Terminale ES](#) [Terminale L](#)

Examens

[Brevet](#) [Bac 2020](#)
[Bac 2021](#)

Matières

[Mathématiques](#) [Français](#) [Histoire](#) [Géographie](#) [Physique Chimie](#) [SVT](#) [SES](#)
[Anglais](#) [Espagnol](#) [Sciences](#)

Offres

[Offre d'accompagnement](#)

Aide

[Contact & FAQ](#)

À propos

[Qui sommes-nous ?](#)