

# Sensibilisation à la didactique de l'informatique (1)

Laurianne Foulquier

[Laurianne.foulquier@u-bordeaux.fr](mailto:Laurianne.foulquier@u-bordeaux.fr)

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

On peut distinguer divers champs d'étude en didactique:

- Une réflexion sur les contenus d'enseignement, en lien étroit avec l'épistémologie.
- Des recherches sur les conditions d'appropriation des savoirs .
- Des recherches en lien avec les pratiques enseignantes...

Le point commun de ces champs non exclusifs les uns aux autres est l'attention aux **savoirs disciplinaires**.

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

- Ainsi, l'épistémologie joue un rôle central dans la constitution même des didactiques des disciplines scientifiques, en permettant de rendre explicites les hypothèses quand à la nature et la structure des savoirs, qui agissent implicitement sur les choix d'enseignement de ces savoirs.
- En fournissant les éléments relatifs à la genèse, la nature, le rôle et la place des savoirs, l'épistémologie contribue de manière significative au travail didactique d'ouverture de choix raisonnés et fondés scientifiquement pour leur enseignement et leur apprentissage. (Modeste)

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

Les quatre concepts de l'informatique de Gilles Dowek :

- L'information
- Les algorithmes
- Les machines
- Les langages



I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

Les quatre concepts de l'informatique de Gilles Dowek :

- L'information
- Les algorithmes
- Les machines
- Les langages

« Il semble plus pertinent d'énumérer, non les sous-disciplines, mais les concepts utilisés dans ce domaine de la connaissance, qui ont sans doute une stabilité temporelle plus grande que les diverses sous-disciplines qui se recomposent sans cesse. Les relations entre ces différents concepts sont également sans doute plus révélatrices de la structure profonde de l'informatique, que les relations entre ces diverses sous-disciplines. » (Dowek)

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

## Préambule programme SNT:

Malgré leur grande variété, ces avancées se fondent toutes sur l'universalité et la flexibilité d'un petit nombre de concepts en interaction :

- les **données**, qui représentent sous une forme numérique unifiée des **informations** très diverses : textes, images, sons, mesures physiques, sommes d'argent, etc. ;
- les **algorithmes**, qui spécifient de façon abstraite et précise des traitements à effectuer sur les données à partir d'opérations élémentaires ;
- les **langages**, qui permettent de traduire les algorithmes abstraits en **programmes** textuels ou graphiques de façon à ce qu'ils soient exécutables par les machines ;
- les **machines**, et leurs systèmes d'exploitation, qui permettent d'exécuter des programmes en enchaînant un grand nombre d'instructions simples, assurant la persistance des données par leur stockage, et de gérer les communications. On y inclut les objets connectés et les réseaux.

À ces concepts s'ajoute un élément transversal : les interfaces qui permettent la communication avec les humains, la collecte des données et la commande des systèmes.

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

## Préambule programme NSI:

L'objectif de cet enseignement, non professionnalisant, est l'appropriation des concepts et des méthodes qui fondent l'informatique, dans ses dimensions scientifiques et techniques. Cet enseignement s'appuie sur l'universalité de quatre concepts fondamentaux et la variété de leurs interactions :

- Les **données**, qui représentent sous une forme numérique unifiée des informations très diverses : textes, images, sons, mesures physiques, sommes d'argent, etc.
- Les **algorithmes**, qui spécifient de façon abstraite et précise des traitements à effectuer sur les données à partir d'opérations élémentaires.
- Les **langages**, qui permettent de traduire les algorithmes abstraits en **programmes** textuels ou graphiques de façon à ce qu'ils soient exécutables par les machines.
- Les **machines**, et leurs systèmes d'exploitation, qui permettent d'exécuter des programmes en enchaînant un grand nombre d'instructions simples, assurent la persistance des données par leur stockage et de gérer les communications. On y inclut les **objets connectés** et les **réseaux**.

À ces concepts s'ajoute un élément transversal : les **interfaces** qui permettent la communication avec les humains, la collecte des données et la commande des systèmes.

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

L'information:

- Représenter des données de manière symbolique et codage de **l'information**
- Lié à la nécessité de représenter l'information sous forme symbolique pour la rendre manipulable par les algorithmes décrits pour les machines (les programmes).
- « La taille du plus court programme qui engendre un message est une manière de définir la quantité d'information contenue dans ce message. Cette théorie de la compression aboutit donc à une théorie quantitative de l'information »
- « L'émergence de cette notion d'information nous a aussi permis de prendre conscience du fait que, dans bien des cas, nous utilisons les machines, non pour transformer de l'information en lui appliquant des algorithmes, mais uniquement pour l'archiver. »

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

Les algorithmes:

- Le concept d'algorithme est le plus ancien, puisque 2 500 ans avant notre ère, les comptables utilisaient déjà des algorithmes pour effectuer les quatre opérations, calculer des prêts, des héritages... et les arpenteurs pour calculer l'aire de surfaces agricoles.
- Un **algorithme** est une « recette » qui permet de résoudre un certain problème de manière systématique (déterministe ou non).
  - Un problème → plusieurs algorithmes
  - Un algorithme → plusieurs programmes.

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

## Les langages

- Décrire un algorithme dans un **langage** de programmation est aujourd'hui indispensable pour que cet algorithme soit exécuté par une machine paramétrable.
- Les premiers langages de programmation universels (lambda-calcul, MT) sont antérieurs de peu aux premières machines universelles. Ils sont liés à la nécessité d'exprimer sans ambiguïté et de communiquer (liens à la logique).
- Il existe des langages formels autres que des langages de programmation : spécifier, raisonner sur des programmes, exprimer des requêtes... → notion très générale de langage (formel) en informatique.

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

## Les machines

- Une **machine** est un système matériel (souvent avec un protocole d'interaction qui permet d'échanger des données).
- « Aujourd'hui, de nombreuses machines exploitent les propriétés physiques des semi-conducteurs, mais rien n'impose que cela soit toujours le cas et la recherche en informatique explore de nombreuses alternatives » (Dowek).
- Ordinateurs : machines universelles qui peuvent exécuter n'importe quel algorithme opérant sur des données symboliques, pourvu qu'on les munisse des bonnes « cartes perforées », du bon programme .
- Après l'ordinateur, le réseau est donc une deuxième instance de ce concept de machine, qui donne à cette question d'extension dans l'espace une place essentielle.

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

« Ces quatre concepts, loin de définir quatre branches de l'informatique, sont souvent utilisés ensemble. C'est parce que nous voulons faire exécuter des algorithmes par des machines paramétrables que nous avons eu besoin d'exprimer ces algorithmes dans des langages formels et les données sur lesquelles ces algorithmes calculent comme des informations. » (Dowek)

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

« Ce souci du caractère algorithmique de la description des objets, du langage dans lequel ces descriptions sont exprimées, des flux d'information et des instruments sont plus généralement caractéristiques d'une 'pensée informatique'. Cette pensée a bien entendu de nombreux précurseurs dans l'histoire, comme Galilée qui a à la fois introduit l'utilisation d'instruments – la lunette – et proposé d'utiliser un nouveau langage – le langage mathématique – dans les sciences de la nature, mais ce qui semble nouveau, en revanche, est de prêter attention simultanément, dans un même geste, à ces quatre questions. » (Dowek)

I. Une discipline informatique

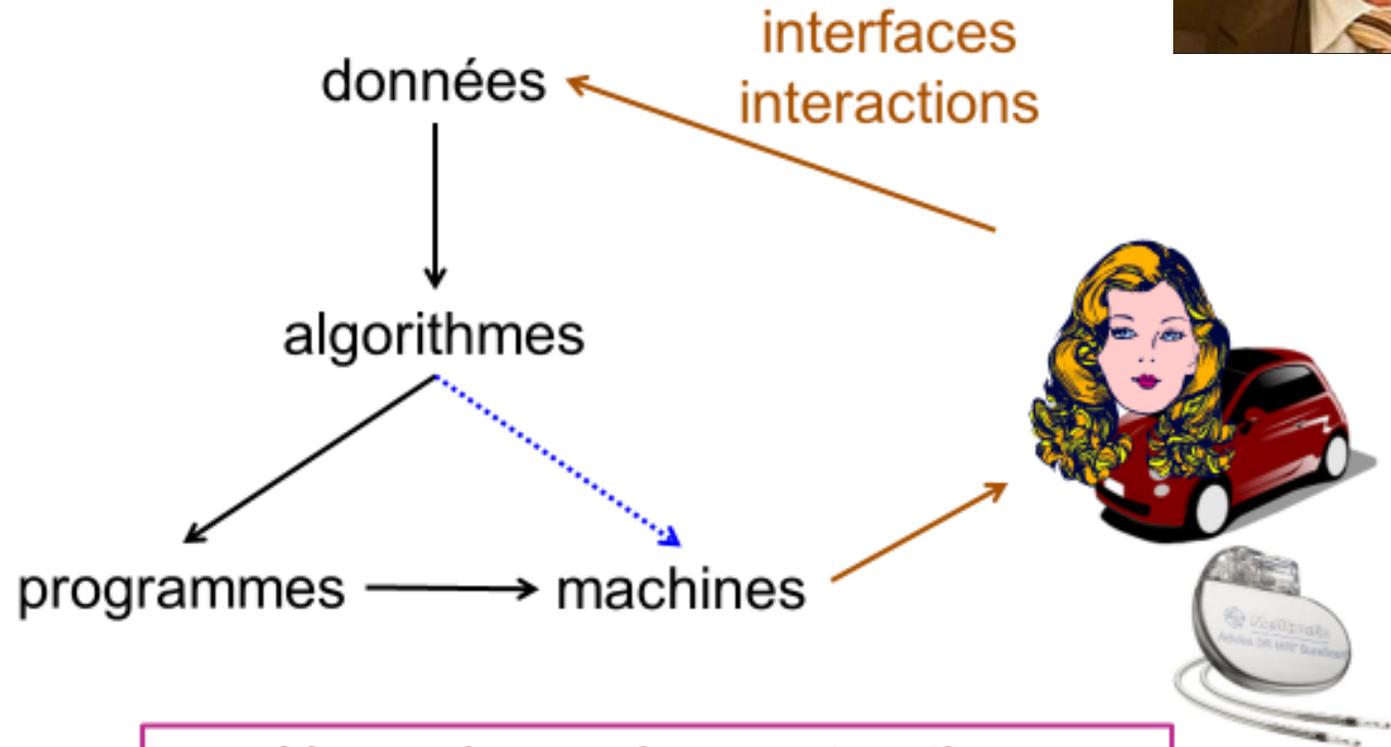
II. Apports didactiques

III. Regard sur trois TP

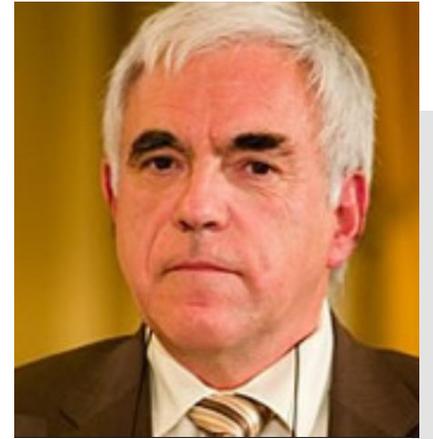
IV. Vers un outil d'aide à la préparation

Gérard Berry les complète par:

## *Les piliers de l'informatique*



Une science de construction,  
très différente des sciences naturelles



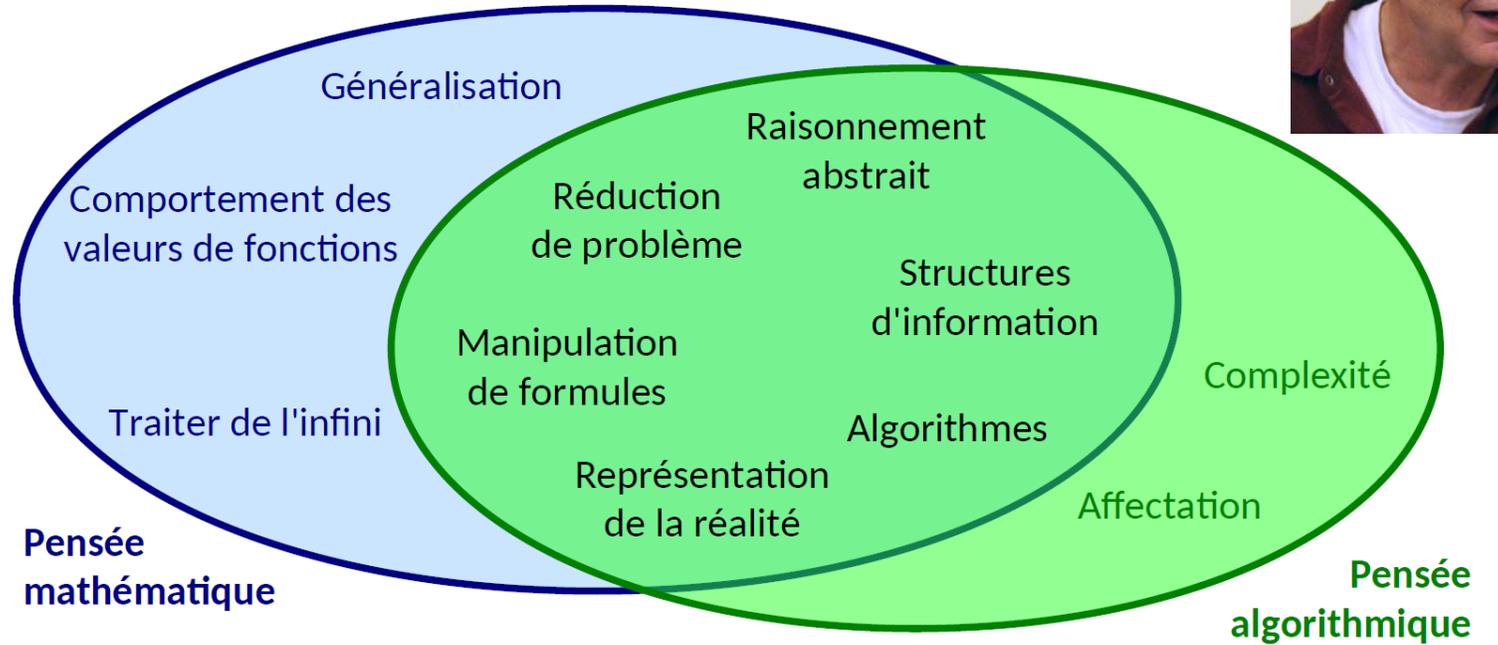
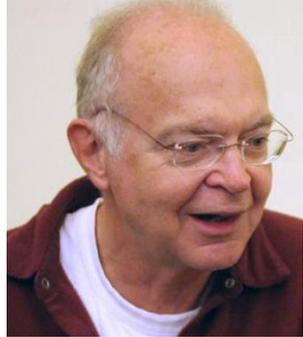
I. Une discipline informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à la préparation

Donald Knuth dans un article de 1985, « algorithmic thinking and mathematical thinking », essaie de différencier pensée mathématique et pensée informatique:



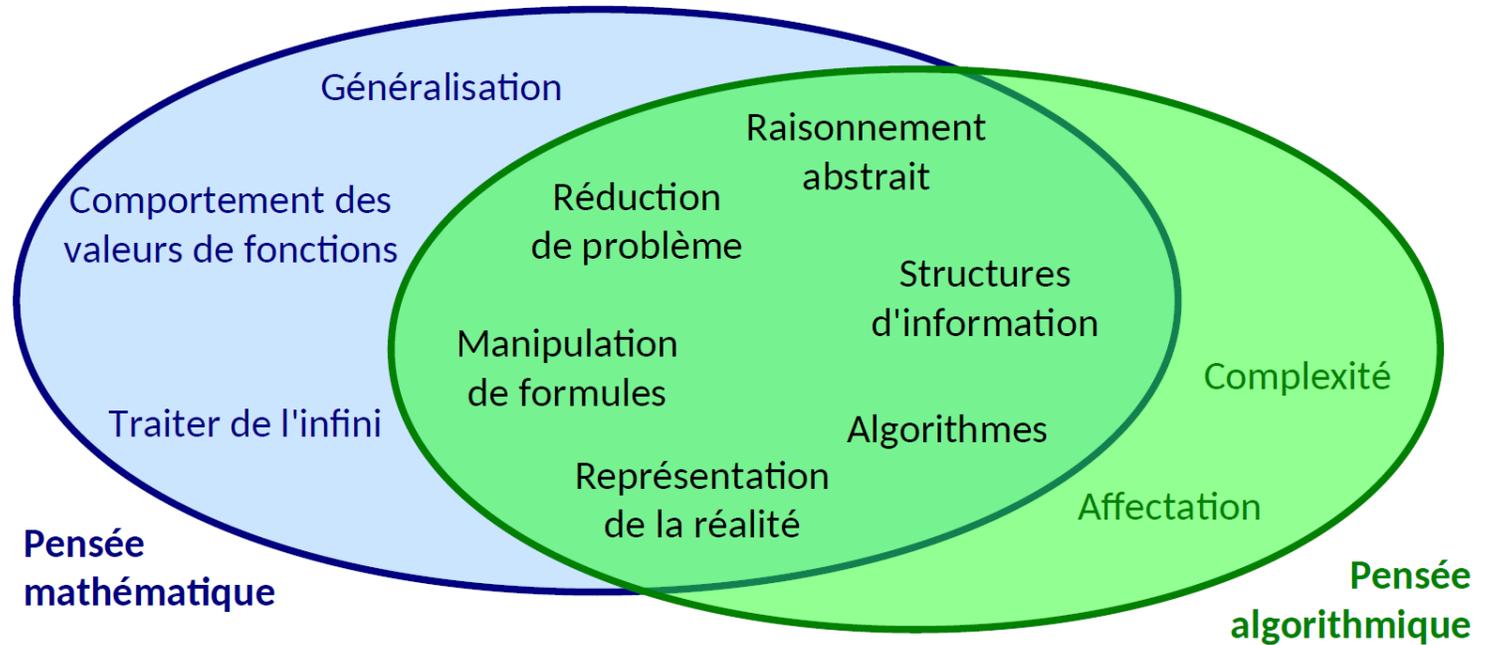
I. Une discipline informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à la préparation

Donald Knuth dans un article de 1985, « algorithmic thinking and mathematical thinking », essaie de différencier pensée mathématique et pensée informatique:



I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

Donald Knuth dans un article de 1985, « algorithmic thinking and mathematical thinking », essaie de différencier pensée mathématique et pensée informatique:

- Complexité comme spécifique de l'activité algorithmique.
- Affectation réfère à la notion de variable informatique.

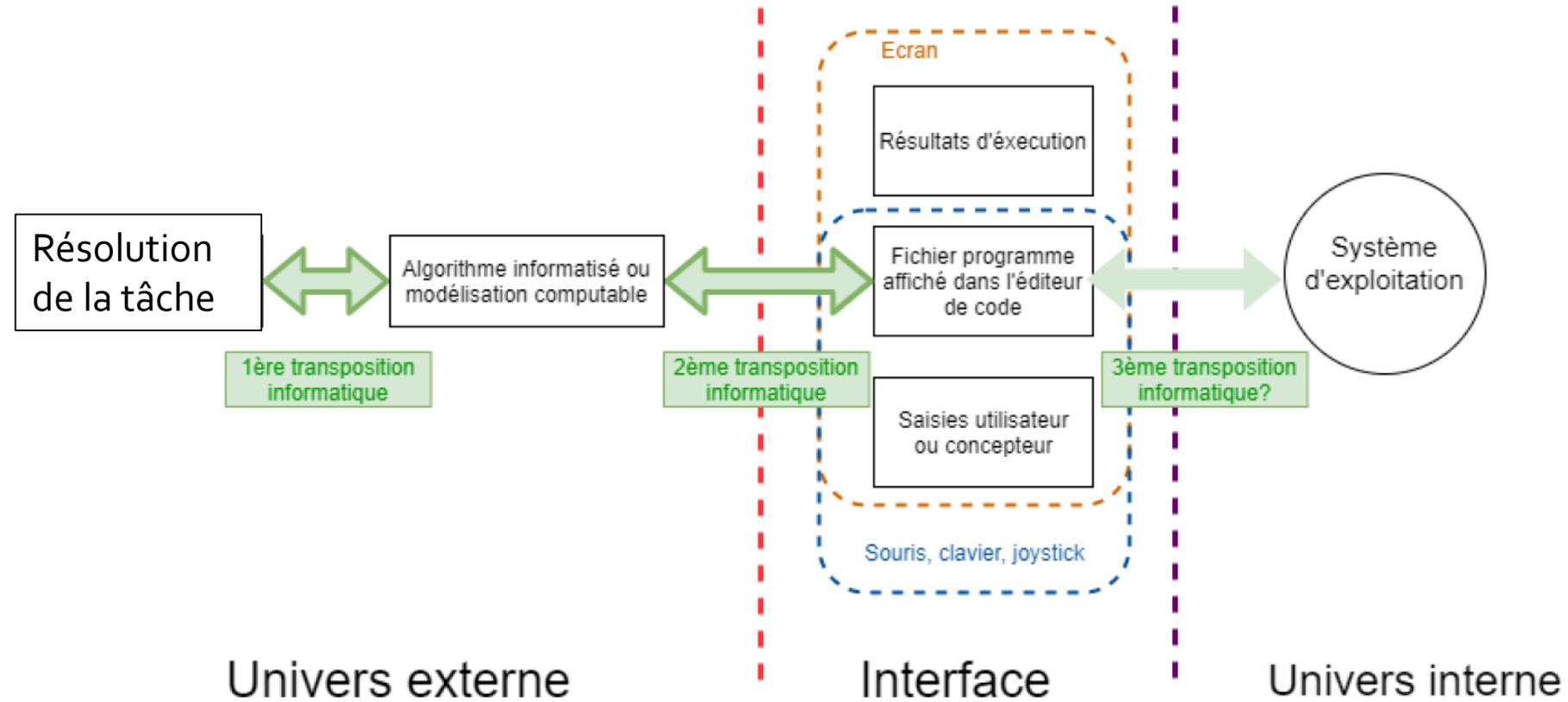
→ Révélateur d'une orientation vers la résolution effective et systématique des problèmes, au-delà de la constructivité mathématique.

I. Une discipline informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à la préparation

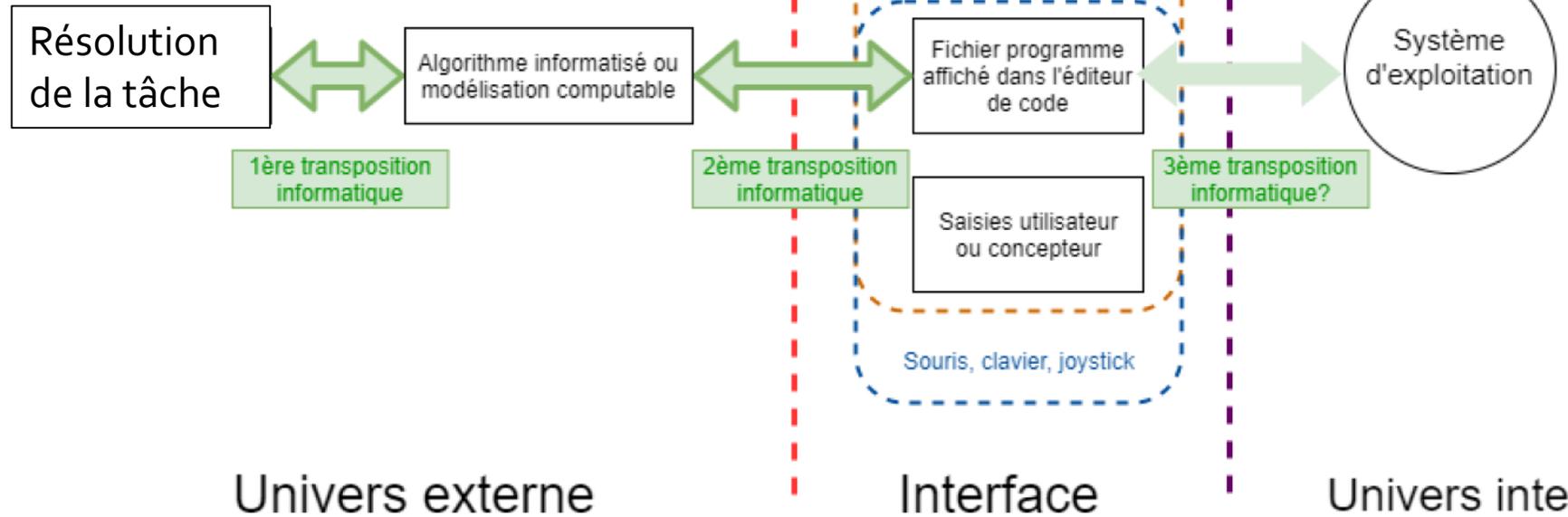
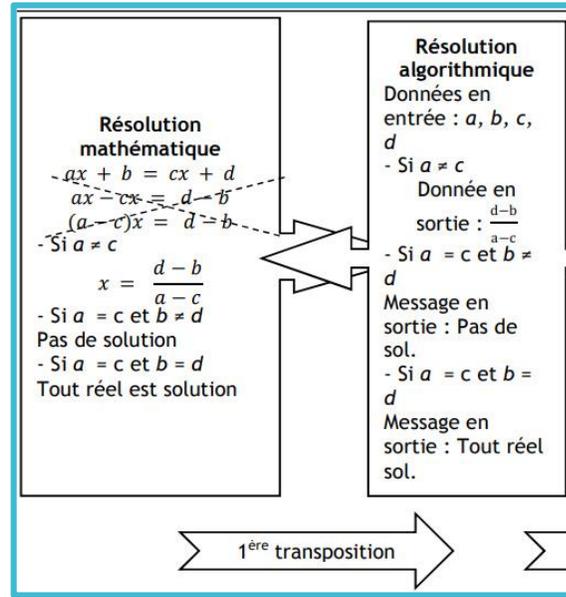


I. Une discipline informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à la préparation



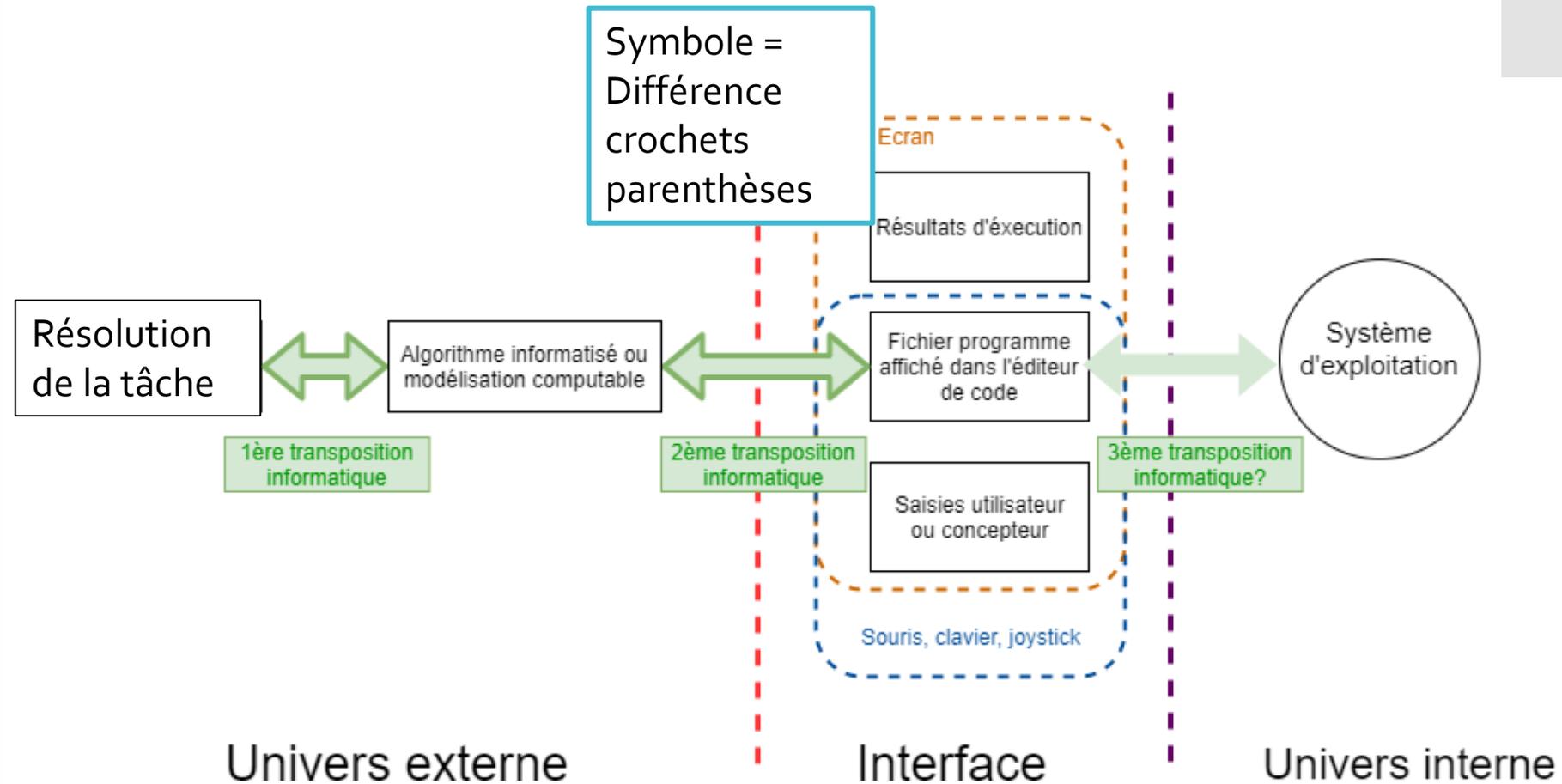
I. Une discipline informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à la préparation

```
def fonction_affine(x):  
    FA=[(x+5)*10-x-10]  
    return(FA)  
  
fonction_affine1 = fonction_affine(10)  
print("fonction_affine x=10 vaut", fonction_affine1)
```



I. Une discipline informatique

II. Apports didactiques

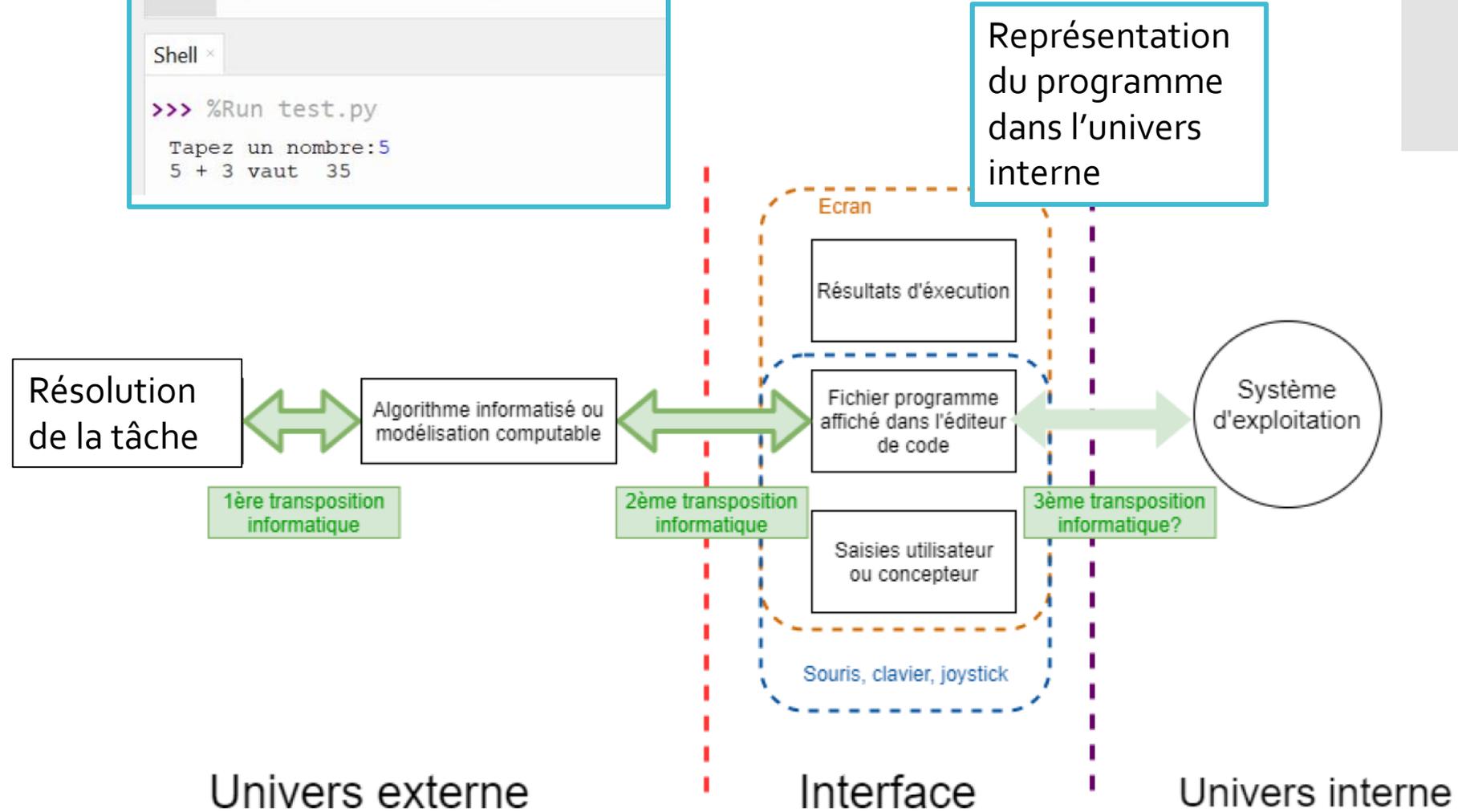
III. Regard sur trois TP

IV. Vers un outil d'aide à la préparation

```
1 a=input("Tapez un nombre:")
2 b="3"
3 c=b+a
4 print(a,"+", b, "vaut ", c)
```

Shell ×

```
>>> %Run test.py
Tapez un nombre:5
5 + 3 vaut 35
```



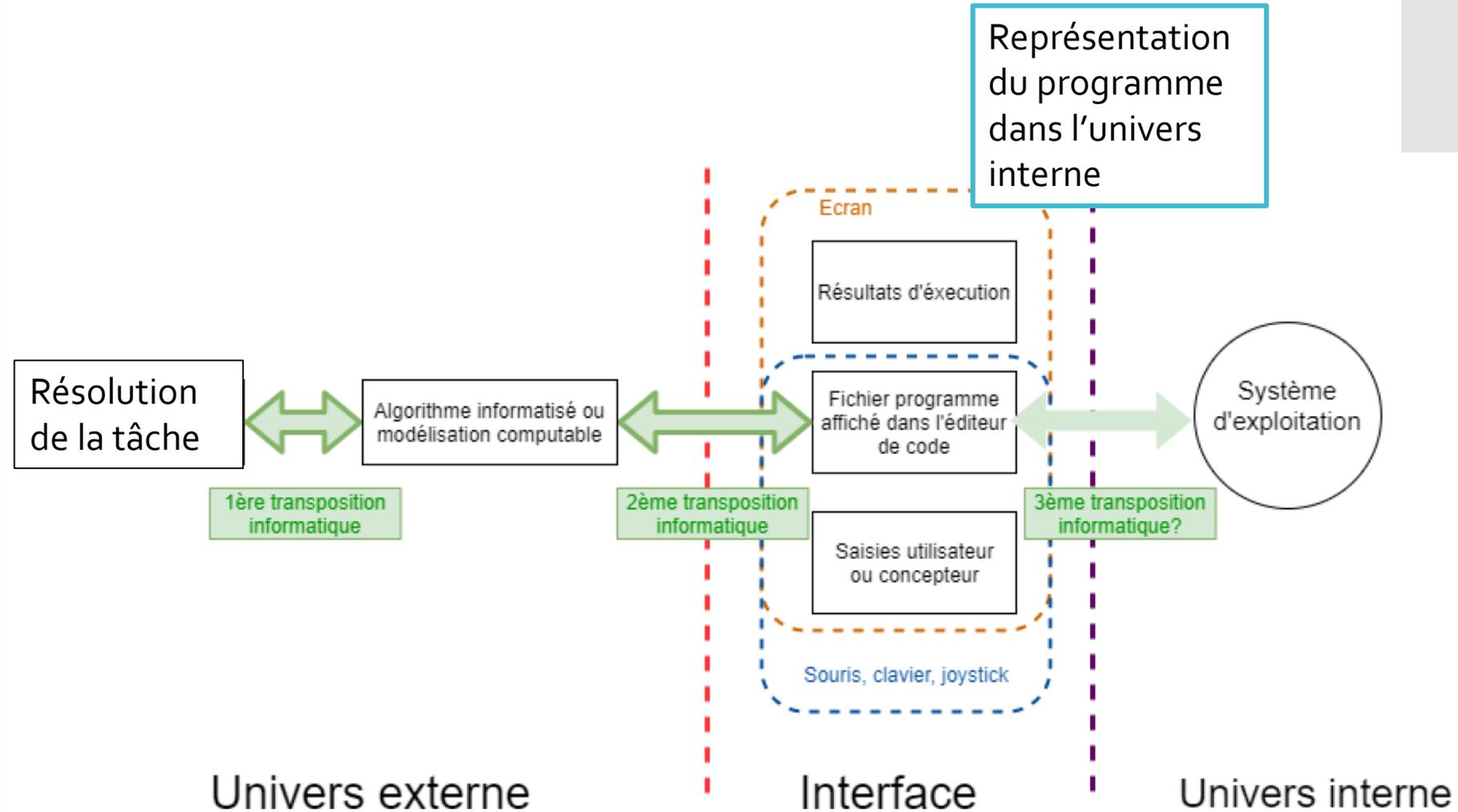
I. Une discipline informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à la préparation

```
7 AFFICHER 'voulez-vous recommencer? (oui/non)'; lire rep
8 SI rep = 'non' ALORS TERMINER SINON ALLER EN 3
```



I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

Pour travailler une notion, si on veut installer une certaine conceptualisation et une organisation des connaissances, il peut être efficace de varier les activités proposées aux élèves.

En particulier, ne pas se limiter à des tâches simples et isolées mais introduire les aspects **outils et objets** et proposer par exemple des problèmes à étapes où ces étapes sont à retrouver.

Travaux de Régine DOUADY

I. Une discipline informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à la préparation

- CONSIGNE: Voici deux TP donnés par des collègues enseignant en ISN trouvés sur le net.

	Objectifs	Points positifs	Points « négatifs »
TP A			
TP B			
TP C			

Exercice 1

- a) Ecrire un programme qui demande à l'utilisateur d'entrer 2 nombres puis qui affiche le plus grand des deux nombres (c'est-à-dire le maximum des deux nombres).
- b) Ecrire un programme qui demande à l'utilisateur d'entrer 2 nombres puis qui affiche le plus petit des deux nombres (c'est-à-dire le minimum des deux nombres).

Exercice 2

Ecrire un programme qui demande à l'utilisateur d'entrer un entier n puis qui calcule la somme des carrés des entiers de 0 à n.

Exercice 3

Ecrire un programme qui demande à l'utilisateur d'entrer 3 nombres puis qui affiche « oui » s'ils sont rangés dans l'ordre croissant et « non » sinon.

Exercice 4

Ecrire un programme qui demande à l'utilisateur d'entrer les coordonnées de deux points A et B dans un repère orthonormé puis qui calcule la distance AB.

Exercice 5

- a) Ecrire un programme qui demande à l'utilisateur d'entrer 3 nombres puis qui les écrit dans l'ordre croissant (tri croissant).
- b) Ecrire un programme qui demande à l'utilisateur d'entrer 3 nombres puis qui les écrit dans l'ordre décroissant (tri décroissant).

Exercice 6

Ecrire un programme qui affiche 4 nombres aléatoires compris entre 0 et 1 et qui calcule la moyenne de ces 4 nombres.

Exercice 7

Ecrire un programme qui affiche 50 nombres aléatoires compris entre 0 et 1 et qui calcule la moyenne de ces 50 nombres.

Exercice 8

A l'aide du module turtle

- a) Ecrire un programme qui trace un rectangle.
- b) Ecrire un programme qui trace un triangle équilatéral en rouge.
- c) Ecrire un programme qui trace un hexagone.
- d) Ecrire un programme qui trace successivement deux triangles équilatéraux successivement qui n'ont pas de point d'intersection.
- e) Ecrire un programme qui trace une étoile à 6 branches.

Exercice 9

Créer une fenêtre graphique simple et y ajouter des widgets.

Vérifier le fonctionnement des différents éléments.

(Livre page 82)



Fonctions (méthode) (langage : JAVA ; Processing).

On appelle **fonction** un sous-programme qui permet d'effectuer un ensemble d'instructions par simple **appel de la fonction dans le programme principal**. Les fonctions permettent d'exécuter dans plusieurs parties du programme une série d'instructions, cela permet une simplicité du code et donc une taille de programme minimale. D'autre part, une fonction peut faire appel à elle-même, on parle alors de fonction réursive.

En Java une fonction est une méthode (Nous verrons plus tard pourquoi cette différence)

Depuis le TP n° 1 nous utilisons des (fonctions) méthodes déjà incluses dans le langage Java.

`println()` est une méthode (fonction) qui permet d'afficher du texte, des résultats...

Les fonctions mathématiques sont des méthodes prédéfinies dans Java : `sin()` ; `sqrt()` ; `exp()` ; `sqrt()` etc.....

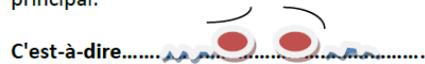
Dans Processing les « instructions » `line()` ; `rect()` ; `ellipse()` ; `fill()` etc.... sont des méthodes propres à Processing.

Nous pouvons également définir nos propres fonctions.....

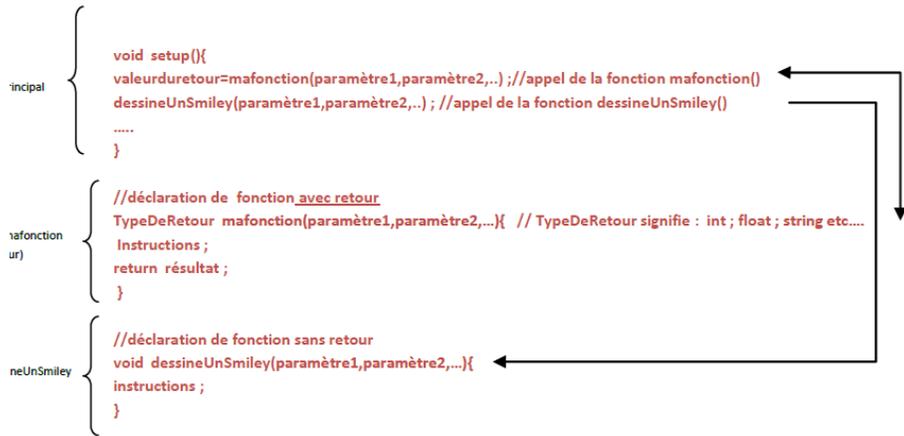
Il y a deux types de fonctions (méthodes) : Celles qui retourne un résultat et celle qui ne retourne rien.

La méthode `line(x1,y1,x2,y2)` ne renvoie aucun résultat , elle dessine une ligne, alors que la méthode `sin(12)` renvoie le résultat du sinus du nombre 12.

Pour pouvoir appeler une méthode dans un programme il faut la déclarer en dehors du programme principal.



C'est-à-dire.....



(En fait une méthode void renvoie le mot-clé void...mais c'est une autre histoire...)

Des exemples pour bien comprendre.....

Exemple 1 : fonction (sans retour) .

Nous allons construire une méthode `dessineUnSmiley(param1,param2)` dans Processing. La tête du smiley est un cercle dont le centre a pour coordonnées ( param1,param2). Un code possible qui dessine un smiley centré en (0,0) est :

```

smooth();
fill(0,255,0);
ellipse(0,0,40,40);
fill(0,0,0);
ellipse(-7,-7,5,5);
fill(0,0,0);
ellipse(7,-7,5,5);
noFill();
bezier(-10,7,-5,12,5,12,10,7);
    
```

Vous pouvez utiliser le code que vous avez écrit au TP n°1

Pour pouvoir le dessiner centré sur un point de coordonnées ( x,y) dans une fenêtre de 200x200, il suffit de le traduire de (x,y).

Essayer le avec `translate(60,150)` ;

Le code suivant appelle la fonction `dessineUnSmiley(60,150)` ;

```

void setup(){
    size(200,200);
    dessineUnSmiley(60,150);
}
    
```



```

void dessineUnSmiley(float x,float y){ // déclaration de la fonction avec des paramètres en float
    translate(x,y);
    smooth();
    fill(0,255,0);
    ellipse(0,0,40,40);
    fill(0,0,0);
    ellipse(-7,-7,5,5);
    fill(0,0,0);
    ellipse(7,-7,5,5);
    noFill();
    bezier(-10,7,-5,12,5,12,10,7);
}
    
```

```

Faire différents essais de coordonnées
Ajouter la méthode
draw(){
    if(mousePressed && (mouseButton==LEFT)){
        dessineUnSmiley(mouseX,mouseY) ;
    }
}
    
```

Modifier le programme précédent pour que : lorsque l'on clique sur le bouton droit de la souris un smiley rouge se dessine et lorsque l'on clique sur le bouton gauche un smiley vert se dessine.

( indication : introduire trois nouveaux paramètres dans la fonction `dessineUnSmiley`)

**Remarque :** L'opérateur logique `&&` dans la condition : `mousePressed && (mouseButton==LEFT)`.

Signifie que la condition `mousePressed` **ET** la condition `mouseButton==LEFT` doivent être vérifiées.

## Exemple 2 : fonction (avec retour) .

Nous allons, dans Processing, créer une fonction qui compare deux entiers et qui nous renvoie le plus grand.

L'algorithme :

< !----Initialisation des variables---->

a et b sont des entiers

max est un entier

< !----Traitement---->

Demander à l'utilisateur les valeurs de a et b

(Utiliser le TP n° 2 pour cela)

Appeler la fonction et mettre le résultat dans max :

max= compare(a,b)

faire afficher le max dans la fenêtre

(voir ci-dessous)

< !---fin du traitement--->



< !----la fonction compare---->

```
int compare(int x,int y){
    if(x>=y){
        return x;
    }
    else{
        return y;
    }
}
```

- ✎ Ecrire ce programme et faire afficher le résultat dans la fenêtre.
- ✎ Concevoir un programme qui compare trois nombres (indication : On pourra appeler plusieurs fois la fonction max...)
- ✎ Concevoir un programme qui trie du plus grand au plus petit trois nombres...



Pour faire afficher dans la fenêtre du texte avec le contenu d'une variable, on procède ainsi :

Le texte est entre guillemet et on sépare le texte de la variable avec un +

Supposons que x vaut 2

```
text("la valeur de x est : "+x,10,10) ;
```

Affichera au point de coordonnées (10,10) : la valeur de x est : 2

# C

## PARCOURS SEQUENTIEL D'UN TABLEAU

Les opérations de traitement informatique d'un volume important de données comme, par exemple, la sélection ou la récupération des données dans une base de données, nécessitent des algorithmes de traitement qui doivent être efficaces.

Les algorithmes de recherche en font partie, leur rôle est de déterminer si une donnée est présente et d'indiquer sa position dans la base, le cas échéant, pour une utilisation annexe. C'est l'un des mécanismes principalement utilisés dans les bases de données afin de récupérer des informations à partir d'un champ (mot clé).

### 1. Les tableaux (rappels)

Un tableau (« *array* » en anglais) est une structure de données de base qui est un ensemble d'éléments du même type, accessibles par leur index.

Il est défini par :

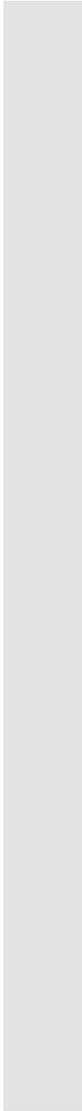
- son **nom** ;
- le **type** de ses éléments ;
- sa **taille** ou le **nombre** de ses éléments.

Le premier index d'un tableau de N éléments est 0 et le dernier index est N – 1 :

0	1	2	3	4	5	6	7	← index
12	14	16	09	11	10	13	17	← données

Un tableau de 8 entiers

→ Les seules opérations possibles sur un tableau sont : la déclaration, l'initialisation (déclaration avec valeurs initiales) et l'accès à ses éléments.



# Vers un outil d'aide à la préparation

I. Une discipline  
informatique

II. Apports didactiques

III. Regard sur trois TP

IV. Vers un outil d'aide à  
la préparation

**Objectifs précis**

**Prérequis**

**Principale situation et analyse a priori :**

En quoi la situation choisie va-t-elle servir l'objectif ?

Quelles sont les pistes de résolution possibles de la situation ?

- Connaissances mises en fonctionnement ?
- Connaissances visées...

Quelles sont les difficultés probables des élèves ?

- En lien avec la connaissance relative des outils en jeu
- En lien avec la résolution de la tâche

**Déroulement chronologique :**

**Principales phases (exemple) :**

**Présentation de l'activité :**

Configuration envisagée (individuel, collectif, groupe) ?

Support envisagé (fichier, document papier, ordinateur ou non à disposition) ?

Rôle de l'enseignant :

S'assure-t-il de la compréhension de la situation et comment ?

S'assure-t-il des différents degrés de connaissance par les élèves de l'environnement numérique et comment ?

Tâche attendue des élèves :

**Phase(s) de recherche :**

Configuration envisagée (individuel, collectif, groupe) ?

Support envisagé (fichier, document papier, ordinateur ou non à disposition) ?

Quelles articulations entre papier / travail sur ordinateur sont prévues ?

Rôle de l'enseignant :

Tâche attendue des élèves :

**Phase(s) d'institutionnalisation :**

Comment organise-t-on la synthèse de la situation ?

Sur quoi porte-t-elle ?

**Prolongements :**

Comment la notion ou le travail du jour sera réinvesti ultérieurement ?

Que prévoir pour les élèves plus rapides ?

**Bilan de la séance :**

Comparaison entre déroulement effectif et prévu (procédures, timing, gestion matérielle, difficultés...)

Et si c'était à refaire, quelles modifications ?

## Objectifs précis

## Prérequis

## Principale situation et analyse a priori :

En quoi la situation choisie va-t-elle servir l'objectif ?

Quelles sont les pistes de résolution possibles de la situation ?

- Connaissances mises en fonctionnement ?
- Connaissances visées...

Quelles sont les difficultés probables des élèves ?

- En lien avec la connaissance relative des outils en jeu
- En lien avec la résolution de la tâche

## Déroulement chronologique :

## Principales phases (exemple) :

### Présentation de l'activité :

Configuration envisagée (individuel, collectif, groupe) ?

Support envisagé (fichier, document papier, ordinateur ou non à disposition) ?

Rôle de l'enseignant :

S'assure-t-il de la compréhension de la situation et comment ?

S'assure-t-il des différents degrés de connaissance par les élèves de l'environnement numérique et comment ?

Tâche attendue des élèves :

### Phase(s) de recherche :

Configuration envisagée (individuel, collectif, groupe) ?

Support envisagé (fichier, document papier, ordinateur ou non à disposition) ?

Quelles articulations entre papier / travail sur ordinateur sont prévues ?

Rôle de l'enseignant :

Tâche attendue des élèves :

### Phase(s) d'institutionnalisation :

Comment organise-t-on la synthèse de la situation ?

Sur quoi porte-t-elle ?

### Prolongements :

Comment la notion ou le travail du jour sera réinvesti ultérieurement ?

Que prévoir pour les élèves plus rapides ?

### Bilan de la séance :

Comparaison entre déroulement effectif et prévu (procédures, timing, gestion matérielle, difficultés...)

Et si c'était à refaire, quelles modifications ?

# Bilan

- Importance de prendre conscience de la discipline informatique dans sa globalité.
- Les technologies numériques sont par exemple de plus en plus conviviales et les élèves d'aujourd'hui, pour la plupart d'entre eux, les fréquentent dès leur plus jeune âge. Pourtant, les artefacts technologiques ne deviennent pour eux des instruments informatiques performants qu'au fil d'un parcours souvent long et qu'il faut savoir accompagner.
- Il faut savoir orchestrer l'usage dans la classe d'une multiplicité d'instruments, à la fois numériques et non numériques.
- Hors de la classe, il faut savoir sélectionner les ressources que l'on va utiliser dans une masse exponentiellement croissante, puis les adapter à son contexte et à ses besoins propres.

# Bibliographie Sitographie

- Conférence de Gilles Dowek, « Premiers principes des langages de programmation »: [https://www.canal-u.tv/video/inria/premiers\\_principes\\_des\\_langages\\_de\\_programmation.6473](https://www.canal-u.tv/video/inria/premiers_principes_des_langages_de_programmation.6473)
- Informatique et sciences du numérique : Édition spéciale Python ! Manuel de spécialité ISN en terminale, Avec des exercices corrigés et des idées de projets, Gilles Dowek, Eyrolles, 2013
- Les quatre concepts de l'informatique, Gilles Dowek, <https://www.epi.asso.fr/revue/articles/a1204g.htm>
- Conférence de Gérard BERRY , « où va l'informatique », Cours au collège de France, <https://www.college-de-france.fr/site/gerard-berry/course-2019-01-23-16hoo.htm>
- Algorithmic thinking and mathematical thinking, D.Knuth, 1988

# Bibliographie Sitographie

- Diaporama de Simon Modeste, DIU Strasbourg
- Etude d'une transposition didactique de l'algorithmique au lycée : une pensée algorithmique comme un versant de la pensée mathématique, N. Briant, Alain Bronner, EMF 2015
- L'informatique et le numérique dans la classe. Qui, quoi, comment?, J.Henry, PUN, 2017
- La théorie des situations didactiques de Brousseau, A.Kuzniack, Repères IREM N°61, Octobre 2005
- Mots-clés de la didactique des sciences, J.P. Astolfi , de Boeck, Mai 2008
- MOOC Efan: <http://www.ens-lyon.fr/savoirs/moocs/moocs-efan>
- Une caméra au fond de la classe de mathématiques, A. Robert, PUFC, 2012
- Une analyse a priori en tryptique comme aide à la création de scénarii d'apprentissage de l'algorithme au lycée, Mémoire Master MEEF, Géraldine Olivier 2021