Concepts de base

Nicolas Delestre
Nicolas.Delestre@insa-rouen.fr
Michel Mainguenaud
Michel.Mainguenaud@insa-rouen.fr

Plan...

- Le formalisme utilisé
- Qu’est ce qu’une variable ?
- Qu’est ce qu’un type ?
- Qu’est ce qu’une expression ?
- Qu’est ce qu’une affectation
- Les entrées / sorties ?

Formalisme...

- Un algorithme doit être lisible et compréhensible par plusieurs personnes
- Il doit donc suivre des règles
- Il est composé d'une entête et d'un corps :
  - l'entête, qui spécifie :
    - le nom de l'algorithme (**Nom :**)
    - son utilité (**Rôle :**)
    - les données "en entrée", c'est-à-dire les éléments qui sont indispensables à son bon fonctionnement (**Entrée :**)
    - les données "en sortie", c'est-à-dire les éléments calculés, produits, par l'algorithme (**Sortie :**)
    - les données locales à l'algorithmique qui lui sont indispensables (**Déclaration :**)

Formalisme...

- le corps, qui est composé :
  - du mot clef **début**
  - d’une suite d’instructions **indentées**
  - du mot clef **fin**

Formalisme...

- Exemple de code :

Nom : addDeuxEntiers

Rôle : Additionner deux entiers a et b et mettre le résultat dans c

Entrée : a,b : entier

Sortie : c : entier

Déclaration : -

début

c ←a+b

fin

Qu’est ce qu’une variable...

- Une variable est une entité qui contient une information :
  - une variable possède un nom, on parle **d’identifiant**
  - une variable possède une valeur
  - une variable possède un type qui caractérise l’ensemble des valeurs que peut prendre la variable
- L’ensemble des variables sont stockées dans la mémoire de l’ordinateur

Qu’est ce qu’une variable...

- On peut faire l’analogie avec une armoire d’archive qui contiendrait des tiroirs étiquetés :
  - l’armoire serait la mémoire de l’ordinateur
  - les tiroirs seraient les variables (l’étiquette corresponderait à l’identidiant)
  - le contenu d’un tiroir serait la valeur de la variable correspondante
  - la couleur du tiroir serait le type de la variable (bleu pour les factures, rouge pour les bons de comande, etc.)

Qu’est ce qu’un type de données...

- Le type d’une variable caractérise :
  - l’ensemble des valeurs que peut prendre la variable
  - l’ensemble des actions que l’on peut effectuer sur une variable
- Lorsqu’une variable apparaît dans l’entête d’un algortithme on lui associe un type en utilisant la syntaxe suivante
  - `Identifiant de la variable :  Son type`
- Par exemple :
  - `age :  Naturel`
  - `nom :  Chaîne de Caractères`
- Une fois qu’un type de données est associé à une variable, cette variable ne peut plus en changer
- Une fois qu’un type de données est associé à une variable le contenu de cette variable doit **obligatoirement** être du même type

Qu’est ce qu’un type de données...

- Par exemple, dans l’exemple précédent on a déclaré `a` et `b` comme des entiers
  - `a` et `b` dans cet algorithme ne pourront pas stocker des réels
  - `a` et `b` dans cet algorithme ne pourront pas changer de type
- Il y a deux grandes catégories de type :
  - les types simples
  - les types complexes (que nous verrons dans la suite du cours)

Les types simples...

- Il y a deux grandes catagories de type simple :
  - Ceux dont le nombres d’éléments est fini, les **dénombrables**
  - Ceux dont le nombre d’éléments est infini, les **indénombrables**

Les types simples dénombrables...

- **booléen**, les variables ne peuvent prendre que les valeurs VRAI ou FAUX
- **intervalle**, les variables ne peuvent prendre que les valeurs entières défi nies dans cet intervalle, par exemple 1..10
- **énuméré**, les variables ne peuvent prendre que les valeurs explicitées, par exemple les jours de la semaine (du lundi au dimanche)
  - Ce sont les seuls types simples qui peuvent être défi nis par l'informaticien
- **caractères**
- Exemples :

```
masculin :  booléen
mois :  1..12
jour :  JoursDeLaSemaine
```

Cas des énumérés...

- Si l'informaticien veut utiliser des énumérés, il doit défi nir le type dans l'entête de l'algorithme en explicitant toutes les valeurs de ce type de la façon suivante :
  - *nom du type = {valeur1, valeur2, ..., valeurn}*
  - Par exemple :
    - JoursDeLaSemaine = {Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi, Dimanche}

Les types simples indénombrables...

- **entier** (positifs et négatifs)
- **naturel** (entiers positifs)
- **réel**
- **chaîne de caractères**, par exemple ’cours’ ou ’algorithmique’
- Exemples :

```
age :  naturel
taille :  réel
nom :  chaîne de caractères
```

Opérateur, opérande et expression...

- Un **opérateur** est un symbole d'opération qui permet d'agir sur des variables ou de faire des "calculs"
- Une **opérande** est une entité (variable, constante ou expression) utilisée par un opérateur
- Une **expression** est une combinaison d'opérateur(s) et d'opérande(s), elle est évaluée durant l'exécution de l'algorithme, et possède une valeur (son interprétation) et un type

Opérateur, opérande et expression...

- Par exemple dans `a+b` :
  - `a` est l'opérande gauche
  - `+` est l'opérateur
  - `b` est l'operande droite
  - `a+b` est appelé une expression
  - Si par exemple `a` vaut 2 et `b` 3, l'expression `a+b` vaut 5
  - Si par exemple `a` et `b` sont des entiers, l'expression `a+b` est un entier

Opérateur...

- Un opérateur peut être unaire ou binaire :
  - **Unaire** s'il n'admet qu'une seule opérande, par exemple l'opérateur non
  - **Binaire** s'il admet deux opérandes, par exemple l'opérateur +
- Un opérateur est associé à *un* type de donnée et ne peut être utilisé qu'avec des variables, des constantes, ou des expressions de *ce* type
  - Par exemple l'opérateur + ne peut être utilisé qu'avec les types arithmétiques (naturel, entier et réel) ou (exclusif) le type chaîne de caractères
  - **On ne peut pas additionner un entier et un caractère**
  - Toutefois *exceptionnellement* dans certains cas on accepte d'utiliser un opérateur avec deux opérandes de types différents, c'est par exemple le cas avec les types arithmétiques (2+3.5)

Opérateur...

- La signifi cation d’un opérateur peut changer en fonction du type des opérandes
  - Par exemple l’opérateur `+` avec des entiers aura pour sens l’addition, mais avec des chaînes de caractères aura pour sens la **concaténation**
    - `2+3` vaut `5`
    - `"bonjour" + " tout le monde"` vaut `"bonjour tout le monde"`

Les opérateurs booléens...

- Pour les booléens nous avons les opérateurs `non`, `et`, `ou`, `ouExclusif`
  - `non`

a	non a
Vrai	Faux
Faux	Vrai

  - `et`

a	b	a et b
Vrai	Vrai	Vrai
Vrai	Faux	Faux
Faux	Vrai	Faux
Faux	Faux	Faux

Les opérateurs booléens...

- `ou`

a	b	a ou b
Vrai	Vrai	Vrai
Vrai	Faux	Vrai
Faux	Vrai	Vrai
Faux	Faux	Faux

- `ouExclusif`

a	b	a ouExclusif b
Vrai	Vrai	Faux
Vrai	Faux	Vrai
Faux	Vrai	Vrai
Faux	Faux	Faux

Les opérateurs sur les énumérés...

- Pour les énumérés nous avons trois opérateurs `succ`, `pred`, `ord` :
  - `succ` permet d'obtenir le successeur, par exemple avec le type `JourDeLaSemaine` :
    - `succ Lundi` vaut `Mardi`
    - `succ Dimanche` vaut `Lundi`
  - `pred` permet d'obtenir le prédecesseur, par exemple avec le type `JourDeLaSemaine` :
    - `pred Mardi` vaut `Lundi`
    - `pred Lundi` vaut `Dimanche`
  - `ord` permet d'obtenir le naturel de l'énuméré spécifi é dans la bijection du type énuméré vers les naturels, par exemple avec le type `JourDeLaSemaine` :
    - `ord Lundi` vaut `0`
    - `ord Dimanche` vaut `6`

Les opérateurs sur les caractères...

- Pour les caractères on retrouve les trois opérateurs des énumérés avec en plus un quatrième opérateur nommé `car` qui est le dual de l'opérateur `ord` avec comme fonction de bijection la table de correspondance de la norme ASCII
  - Cf. http://www.commentcamarche.net/base/ascii.htm
  - Par exemple
    - `ord A` vaut `65`
    - `car 65` vaut `A`
    - `pred A` vaut `@`

Les opérateurs sur les naturels, entiers et réels...

- On retrouve tout naturellemment `+`, `-`, `/`, `*`
- Avec en plus pour les naturels et les entiers `div` et `mod`, qui permettent respectivement de calculer une division entière et le reste de cette division, par exemple :
  - `11 div 2` vaut `5`
  - `11 mod 2` vaut `1`

L’opérateur pour les chaînes de caractères...

- C’est l’opérateur de concaténation vu précédemment qui est +

L’opérateur d’égalité, d’inégalité, etc...

- L’opérateur d’égalité
  - C’est l’opérateur que l’on retrouve chez tous les types simples qui permet de savoir si les deux opérandes sont égales
  - Cet opérateur est représenté par le caractère $=$
  - Une expression contenant cet opérateur est un booléen
- On a aussi l’opérateur d’inégalité $\neq$
- Et pour les types possédant un ordre les opérateurs de comparaison $<, \leq, \geq, >$

Priorité des opérateurs...

- Tout comme en arithmétique les opérateurs ont des priorités
  - Par exemple * et / sont prioritaires sur + et -
- Pour les booléens, la priorité des opérateurs est `non`, `et`, `ouExclusif` et `ou`
- Pour clarifi er les choses (ou pour dans certains cas supprimer toutes ambiguités) on peut utiliser des parenthèses

Actions sur les variables...

- On ne peut faire que deux choses avec une variable :
  1. Obtenir son contenu (*regarder le contenu du tiroir*)
     - Cela s'effectue simplement en nommant la variable
  2. Affecter un (nouveau) contenu (mettre une (nouvelle) information dans le tiroir)
     - Cela s'effectue en utilisant l'opérateur d'affectation représenter par le symbole ←
     - La syntaxe de cet opérateur est :
       - `identifiant de la variable` ← `expression sans operateur d'affectation`

Actions sur les variables...

- Par exemple l'expression `c ← a + b` se comprend de la façon suivante :
  - On prend la valeur contenue dans la variable `a`
  - On prend la valeur contenue dans la variable `b`
  - On additionne ces deux valeurs
  - On met ce résultat dans la variable `c`
  - Si `c` avait auparavant une valeur, cette dernière est perdue !

Les entrées/sorties...

- Un algorithme peut avoir des interactions avec l'utilisateur
- Il peut afficher un résultat (du texte ou le contenu d'une variable) et demander à l'utilisateur de saisir une information afin de la stocker dans une variable
- En tant qu'informaticien on raisonne en se mettant **"à la place de la machine"**, donc :
  - Pour affi cher une information on utilise la commande **écrire** suivie entre parenthèses de la chaîne de caractères entre guillemets et/ou des variables de type simple à affi cher séparées par des virgules, par exemple :
    - **écrire**("Le valeur de la variable a est", a)
  - Pour donner la possibilité à l'utilisateur de saisir une information on utilise la commande **lire** suivie entre parenthèses de la variable de type simple qui va recevoir la valeur saisie par l'utilisateur, par exemple :
    - **lire**(b)

Exemple d’algorithme...

Nom : euroVersFranc1

Rôle : Convertisseur des sommes en euros vers le franc, avec saisie de la somme en euro et affi chage de la somme en franc

Entrée : -

Sortie : -

Déclaration : valeurEnEuro,valeurEnFranc,tauxConversion : Réel

début

```
    tauxConversion ←6.55957
    écrire("Votre valeur en euro :")
    lire(valeurEnEuro)
    valeurEnFranc ←valeurEnEuro * tauxConversion
    écrire(valeurEnEuro," euros = ",valeurEnFranc," Frs")
```

fin

Exemple d’algorithme...

Nom : euroVersFranc2

Rôle : Convertisseur des sommes en euros vers le franc

Entrée : valeurEnEuro : Réel

Sortie : valeurEnFranc : Réel

Déclaration : tauxConversion : Réel

début

tauxConversion ←6.55957

valeurEnFranc ←valeurEnEuro * tauxConversion

fin