|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Section  Date  Enseignant  Matière | **B1** |  |  |
| **M. MALDONADO**  **[EIAD101] ALGORITMIQUE** | | |

**VARIABLES ET TYPES**

### Variables

**Préliminaires**

* Rappel :   
   Données ⇒ Programme ⇒ Résultats  
    
   données et résultats stockés en mémoire centrale

**Définition**

* Déf 1 : étiquette servant à repérer un emplacement en MC
* Déf 2 : donnée ou objet manipulé par un algorithme (et les langages de programmation)

**Caractéristiques**

* **Nom** (identificateur)  
  - suite de caractères alphanumérique  
  aucune restriction en algorithmique (pas d’espace)  
  contraintes en programmation (langage) :   
  *caractères accentués, commence par une lettre, case-sensitive*, …  
    
  - **Unique**  
    
  - Règles de **nommage** (**conventions**)  
   jourdelasemaine  
   enumjourdelasemaine   
   jour\_de\_la\_semaine  
   jourDeLaSemaine  
   JourDeLaSemaine  
    
  **- lisibilité** (auto-descriptif)
* **Type  
  - Ensemble des valeurs** que peut prendre la variable  
  **- Types** **standard / type construits (utilisateurs)**  
  - Correspondance : **type** ⇒ **taille** mémoire  
   **type** ⇒ **opérations autorisées**
* **Valeur**

**- contenu** de la case mémoire  
**- valeur parasite** initialement  
 ⇒

**Opérations**

* **Déclaration**  
  (*Algo*) Var : jourDeLaSemaine : chaîne de caractère  
   nbEssais : entier  
    
  (C#) string jourDeLaSemaine ;  
   int nbEssais ;  
    
  (*VB.NET / VBA*) dim jourDeLaSemaine as string  
   dim nbEssais as integer  
    
  (*Python*) # pas de déclaration (première affectation)
* **Ecriture**  
  ⇒ valeur précédente écrasée
* Première écriture : **Initialisation**
* **Lecture**

### 2. Types

**Définition**

* Liste des valeurs que peut prendre une variable

**Types simples (prédéfinis)**

* **Entier**  
  12, -5, 0  
  distinction possible : entier court / entier long
* **Réel**3.14, 1.2E5, …
* **Caractère**  
  une touche du clavier  
  ‘A’, ‘\*’, …
* **Chaîne de caractères**  
  plusieurs caractères  
  taille
* **Booléen**   
  deux états : vrai / faux (true / false)

**Types élaborés (construits)**

* **Tableaux**   
  une dimension, deux dimensions (matrice), …
* **Pointeurs**
* **Structures**nom + prénom + ddn + tel + mail
* **Classes**nom + prénom + ddn + tel + mail + CalculerAge + EnvoyerMail  
  (approche **objet**)

**Types valeur vs types références**

* **Valeur**  
  (dessin)  
  *C# : booléen, types numériques, structures*
* **Référence**  
  (dessin)  
  *C# : chaine, objet*

**Contrôles de types**

* **Langages pas / faiblement typés**  
  Aucun ou peu de contrôles – Langages permissifs  
  *C, Php, JavaScript*
* **Langages fortements typés**  
  Contrôles très stricts :   
  Détection des erreurs de typage à la compilation  
  *ADA, JAVA, C#, Python*
* **Concordance de type**  
  *numérique vs chaîne, entier vs réel, …*
* **Concordance de taille***entier court vs entier long (perte de précision)*
* **Typage statique / dynamique**compilateur / exécution  
  *C# (statique) / Python (dynamique)*

**Conversions**

* **Transformation d’un type vers un autre**  
  chaînes de conversion   
  entier court ⇒ entier long ⇒ flottant simple précision ⇒ flottant double précision
* **Implicites**réalisées par le compilateur  
  Exemple : entier ⇒ booléen (C) int i = 3.14 ;  
   if (i) …
* **Explicites (*cast*)**forcées par le programmeur  
    
   (C) float f = 3.14 ;  
   int i = (int) f ;  
    
   (C#) float f = 3.14 ;  
   int i = f as int;  
    
   (Python) f = 3.14  
   i = int (f)

### 3. Affectation

**Définition**

* **Ranger une valeur** dans une **case mémoire**
* **Une seule valeur**  
  valeur précédente écrasée
* **Souvent oublié**Source d’erreur **-**une erreur de programmation sur 2 (C)

**Initialisation**

* **Première affectation**
* **Remarque :**  
  C# : variables **numériques** initialisées à **0**  
   variables **références** initialisées à **null**

**Valeur gauche (left value)**

* **Expression affectable**  
  exemple : nom de variable  
    
  *variable = { valeur , expression, variable }*