Les principales opérations qui permettent de former les expressions arithmétiques sont `+`, `-`, `*`, `/` et `%`, ces deux dernières opérations étant respectivement les quotient et reste de la division euclidienne.

Quand l'un des nombres `a` ou `b` est négatif, le nombre `a / b` est le quotient « arrondi vers 0 », c'est-à-dire, le nombre dont la valeur absolue est le quotient des valeurs absolues de `a` et `b`, qui est positif quand `a` et `b` sont de même signe et négatif sinon. Le nombre `a % b` est `a - b * (a / b)`. Ainsi `(-29) / 4` est égal à `-7` et `(-29) % 4` à `-1`.

Les opérations sur les nombres à virgule sont `+`, `-`, `*`, `/` et quelques fonctions transcendantes : `Math.sin`, `Math.cos`, ...

Les principales opérations qui permettent de former des expressions booléennes sont `==`, `!=` — différent —, `<`, `>`, `<=`, `>=`, `&` — et —, `&&`, `|` — ou —, `||` et `!` — non.

Sur tous les types l'expression `(b) ? t : u` a la même valeur que `t` si l'expression booléenne `b` a la valeur `true` et a la même valeur que `u` si l'expression booléenne `b` a la valeur `false`.

Tab. 1.2 Les expressions en Java

Les chaînes de caractères sont de type `String`. Les constantes s'écrivent entre guillemets, par exemple `"École polytechnique"`.

Tab. 1.3 Les chaînes de caractères en Java

Pour déclarer une variable d'un type `T`, on doit remplacer le type `int` par `T`. La forme générale d'une déclaration est donc `{T x = t ; p}`.

En Caml, la déclaration d'une variable s'écrit `let x = ref t in p` *et il n'est pas nécessaire d'indiquer explicitement le type de la variable. Il n'est pas possible, en Caml, de déclarer une variable sans lui donner de valeur initiale.*

En C, comme en Java, la déclaration s'écrit `{T x = t ; p}`*. Il est possible de déclarer une variable sans lui donner de valeur initiale, et dans ce cas, elle peut avoir n'importe quelle valeur.*

En Java et en C, il est impossible de déclarer deux fois la même variable et le programme suivant est incorrect.

```
int y = 4;
int x = 5;
int x = 6;
```