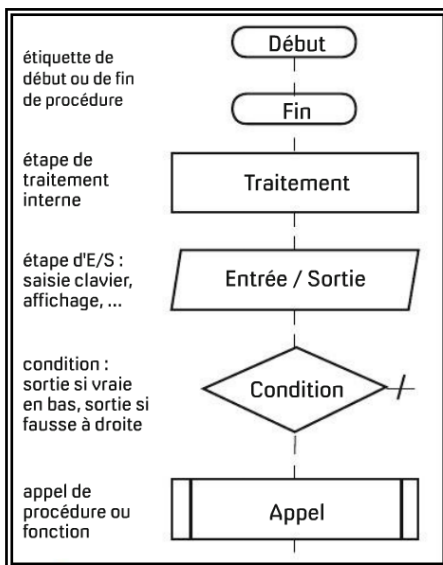
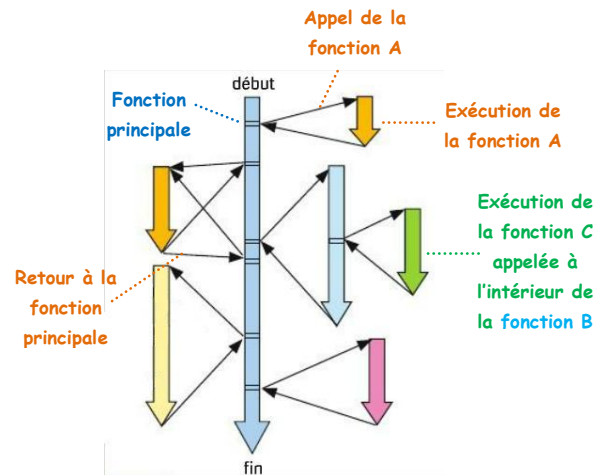


1. Programmation procédurale

La programmation procédurale est basée sur le concept d'appel de procédures (appelées également fonction). Une fonction contient une série d'actions à réaliser. Toute fonction peut être appelée à n'importe quelle étape du fil d'exécution d'un programme, donc aussi à l'intérieur d'une autre fonction.



2. Organigramme

C'est un mode de représentation graphique constitué de symboles tels que rectangles, losanges et flèches. Il a pour avantage d'être très visuel. Dans la pratique 5 symboles suffisent à représenter un algorithme (on parle alors d'algorithme). Les algorithmes décrivent des actions et les chemins à parcourir entre ces actions.

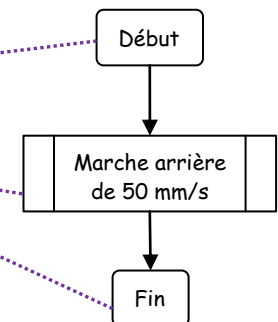
3. Exemple dans le cas de la commande du Robotino

actionPerformed() est une fonction appelée lorsque l'utilisateur clique sur le bouton marche arrière du pavé de commande.

setVelocity() est alors appelé par la fonction *actionPerformed()*. Son exécution permet de commander Robotino en marche arrière.

```
public class ButtonDownListener implements ActionListener {
    public void actionPerformed(ActionEvent arg0) {
        setVelocity(-50, 0, 0);
    }
}
```

Les paramètres de la fonction *setVelocity()* permettent de préciser la valeur de la vitesse linéaire et de rotation du Robot.



Algorithme de la fonction *actionPerformed()*

4. Intérêt des fonctions

Les fonctions permettent d'écrire de manière compacte des programmes et d'en augmenter la lisibilité à condition de donner un nom explicite aux fonctions. De plus le code qui correspond à la fonction n'est stocké qu'une seule fois en mémoire ce qui permet d'optimiser l'espace mémoire utilisé par les programmes. Tout programme commence par l'appel d'une fonction que l'on nomme fonction principale. La fonction principale peut appeler par la suite d'autres fonctions.

