### Liste des fonctions

- ♣ Fonction 1 : setPosition
- ♣ Fonction 2 : cercle
- ♣ Fonction 3 : demiCercle
- ♣ Fonction 4 : carre
- ♣ Fonction 5 : triangle
- ♣ Fonction 6 : rectangle
- Fonction 7 : polygone
- ♣ Fonction 8 : trapeze
- Fonction 9 : losange
- Fonction 10 : semiEllipse
- ♣ Fonction 11 : ellipse
- Fonction 12 : support
- ♣ Fonction 13 : simpleWindow
- Fonction 14: shortWindow
- ♣ Fonction 15 : longWindow
- ♣ Fonction 16 : windowSpace

## Commentaires de spécification

- **Fonction 1**: setPosition
  - Objectifs: Placer la tortue à une position donnée sans tracer
  - o *Méthode* : usage de fonctions du module Turtle
  - o **Besoins**: abscisse, ordonnee
  - o Connus: -
  - o Entrées : abscisse, ordonnee
  - o Sorties: -
  - o *Résultats* : la position de la tortue, modifiée
  - o Hypothèses: -
- **Fonction 2** : cercle
  - o Objectifs: tracer un cercle
  - o *Méthode* : usage de fonctions du module Turtle
  - o **Besoins**: rayon
  - o Connus: -
  - o Entrées : rayon
  - o Sorties: -
  - o **Résultats**: le cercle de rayon 'rayon' tracé sur <u>Python Turtle Graphics</u>
  - o  $Hypoth\`eses$ : rayon > 0
- **4** Fonction 3 : demiCercle

- o *Objectifs*: tracer un demi-cercle
- o *Méthode* : usage de fonctions du module Turtle
- o Besoins: rayon
- o Connus: -
- o Entrées : rayon
- o Sorties: -
- o **Résultats** : le demi-cercle de rayon 'rayon' tracé sur <u>Python Turtle Graphics</u>
- o  $Hypoth\`eses$ : rayon > 0

#### **Fonction 4** : carre

- o Objectifs: tracer un carré
- o *Méthode* : usage de structure itérative et de fonctions du module Turtle
- o *Besoins* : cote, positionFinale
- o Connus: -
- o Entrées : cote
- o Sorties:-
- o **Résultats** : le carre de côté 'cote' tracé sur Python Turtle Graphics
- o  $Hypoth\`eses : cote > 0$

#### **Fonction 5**: triangle

- o *Objectifs*: tracer un triangle
- o *Méthode* : usage de fonctions du module Turtle
- o **Besoins**: base, coteDroit, angleDroit, positionInitiale
- o Connus: -
- o *Entrées* : base, coteDroit, angleDroit
- o Sorties: -
- o **Résultats** : le triangle tracé sur <u>Python Turtle Graphics</u>
- Hypothèses: base > 0, coteDroit > 0, angleDroit > 0 and angleDroit < 180

#### **Fonction 6**: rectangle

- o *Objectifs*: tracer un rectangle
- o Méthode: usage de fonctions du module Turtle
- o **Besoins**: longueur, largeur, positionFinale
- o Connus: -
- o *Entrées* : longueur, largeur
- o Sorties: -
- o **Résultats** : le rectangle de longueur 'longueur' et de largeur 'largeur' tracé sur *Python Turtle Graphics*
- o  $Hypoth\`eses$ : longueur > 0, largeur > 0

#### **Fonction 7**: polygone

- o *Objectifs*: tracer un polygone
- o *Méthode* : usage de fonctions du module Turtle dans une structure itérative
- o **Besoins**: nombreCotes, normeCote, angle, i

- o Connus: -
- o *Entrées* : nombreCotes, normeCote
- o Sorties: -
- Résultats : le polygone de cote 'normeCote' tracé sur <u>Python Turtle</u>
  <u>Graphics</u>
- *Hypothèses*: normeCote > 0, nombreCotes >= 5

#### **Fonction 8**: trapeze

- o *Objectifs*: tracer un trapèze
- o Méthode: usage de fonctions du module Turtle
- Besoins: grandeBase, petiteBase, coteDroit, angleBasDroit, positionInitiale, positionFinale
- o Connus: -
- o Entrées : grandeBase, petiteBase, coteDroit, angleBasDroit
- o Sorties:-
- o **Résultats** : le trapeze tracé sur Python Turtle Graphics
- Hypothèses: grandeBase > 0, petiteBase > 0, coteDroit > 0, angleBasDroit > 0 and angleBasDroit < 180</li>

#### **Fonction 9** : losange

- o *Objectifs*: tracer un losange
- o *Méthode* : usage de structure itérative et de fonctions des modules Turtle et Math
- o Besoins: diagonaleVerticale, diagonaleHorizontale, angle, pi, cote
- o Connus: pi
- o *Entrées* : diagonaleVerticale, diagonaleHorizontale
- o Sorties:-
- o Résultats : le losange tracé sur Python Turtle Graphics
- o *Hypothèses*: diagonaleVerticale > 0, diagonaleHorizontale > 0

### **Fonction 10**: semiEllipse

- Objectifs: tracer la partie supérieure d'une ellipse par rapport à l'axe horizontale
- o *Méthode* : usage de fonctions du module Turtle
- o **Besoins**: rayon
- o Connus: -
- o *Entrées* : rayon
- o Sorties: -
- Résultats : la partie supérieure de l'ellipse de rayon 'rayon' tracé sur <u>Python</u>
  Turtle Graphics
- o  $Hypoth\`eses$ : rayon > 0

#### **4** Fonction 11 : ellipse

- o *Objectifs*: tracer une ellipse
- o *Méthode* : appel de fonction et usage de structure itérative
- o Besoins: rayon, i
- o Connus: -
- o Entrées : rayon
- o Sorties: -
- o *Résultats* : l'ellipse de rayon 'rayon' tracé sur *Python Turtle Graphics*
- o  $Hypoth\`eses$ : rayon > 0

#### **Fonction 12** : support

- o Objectifs: tracer un support du pont
- o *Méthode* : usage de fonctions du module Turtle
- o *Besoins* : a, b, c, d, e, f, g
- o Connus: -
- o Entrées: -
- o Sorties:-
- o *Résultats* : un support du pont tracé sur *Python Turtle Graphics*
- o Hypothèses: -

#### **Fonction 13** : simpleWindow

- o *Objectifs*: tracer une fenêtre simple
- o *Méthode* : appel de fonction et usage de fonctions du module Turtle
- o Besoins: positionActuelle
- o Connus: -
- o Entrées: -
- o Sorties: -
- o **Résultats** : une fenêtre simple tracée sur <u>Python Turtle Graphics</u>
- o Hypothèses:-

#### **Fonction 14**: shortWindow

- o *Objectifs*: tracer une fenêtre carrée à quatre parties
- o *Méthode* : appel de fonction et usage de fonctions du module Turtle
- o Besoins : -
- o Connus: -
- o Entrées: -
- o Sorties: -
- o **Résultats** : la fenêtre carrée tracée sur Python Turtle Graphics
- o Hypothèses: -

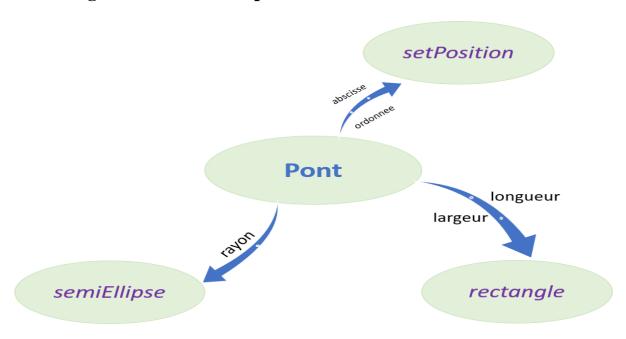
### **Fonction 15** : longWindow

o *Objectifs*: tracer une fenêtre longue à quatre parties

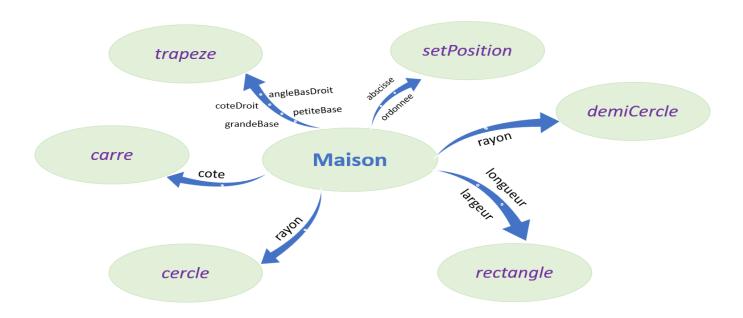
- o *Méthode*: appel de fonction et usage de fonctions du module Turtle
- o Besoins: -
- o Connus: -
- o Entrées: -
- o Sorties: -
- o **Résultats** : la fenêtre longue tracée sur <u>Python Turtle Graphics</u>
- o Hypothèses: -
- **Fonction 16**: windowSpace
  - o *Objectifs*: tracer un espace avec fenêtres longue et courte
  - o *Méthode* : appel de fonction et usage de fonctions du module Turtle
  - o **Besoins**: positionActuelle
  - o Connus: -
  - o Entrées : -
  - o Sorties:-
  - o Résultats: un espace pour fenêtres tracé sur Python Turtle Graphics
  - o Hypothèses: -

## Diagrammes de flux

### **❖** Diagramme de flux du pont



## ❖ Diagramme de flux de la maison



# Tableau de flux

## **❖** Tableau de flux du pont

Pont	Fournit (entrées)	Recoit (sorties)
setPosition	abscisse, ordonnee	-
semiEllipse	rayon	-
Rectangle	longueur, largeur	-

### **❖** Tableau de flux de la maison

Maison	Fournit (entrées)	Recoit (sorties)
setPosition	abscisse, ordonnee	-
demiCercle	rayon	-
cercle	rayon	-
carre	cote	-
rectangle	longueur, largeur	-
trapeze	grandeBase, petiteBase,	-
_	coteDroit, angleBasDroit	