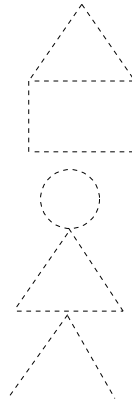


Exemple

Dessiner une maison et une fillette

1 Énoncé du problème (Spécification)

Écrire un programme qui dessine une maison puis la silhouette d'une fillette sur un terminal, comme illustrer ci-dessous :



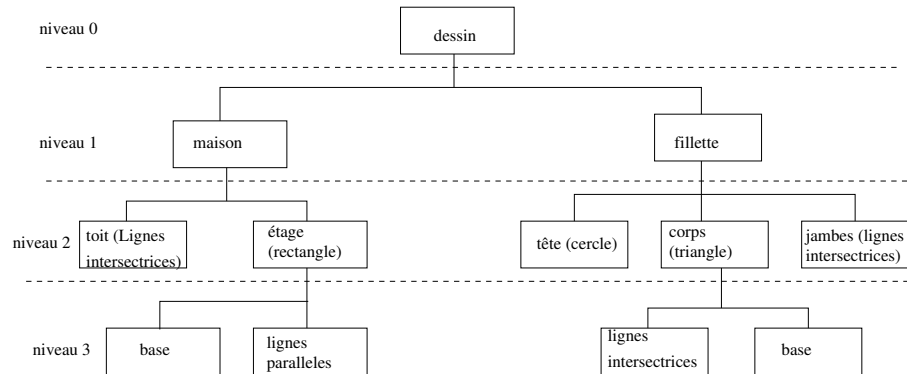
2 Analyse globale

- *Entrée* : Aucune donnée en entrée.
- *Sortie* : Les figures se décomposées de la façon suivante :
 - maison - un triangle sans base et un rectangle.
 - fillette - un cercle, un triangle et un triangle sans base.

On a donc besoin d'un cercle, d'un triangle, d'un rectangle et d'un triangle sans base. Un rectangle et un triangle peuvent se décomposer en deux lignes verticales parallèles et deux «bases» et de deux lignes à intersection et une base. On doit donc imprimer un triangle sans base, deux ligne verticales parallèles, un ligne (base) et un cercle (ces dernières sont les fonctions de base).

- *Formule* : Aucune
- *Constante* : Aucune (sauf le traits pour les dessins);
- *Contrainte* : Aucune
- *Erreur* : Aucune

3 Conception globale (diagramme structurel)



Exemple : Dessiner une maison et une fillette (suite)

4 Conception détaillée module par module

4.1 Module Dessin (principal)

- Analyse
 - Entrée : Aucune
 - Sortie : Aucune
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner la maison (module *maison*) ;
 2. Dessiner la fillette (module *fillette*).

4.2 Module maison

- Analyse
 - Entrée : Aucune
 - Sortie : Aucune
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner le toit (module *ligne intersectrices*) ;
 2. Dessiner la base (module *rectangle*).

4.3 Module fillette

- Analyse
 - Entrée : Aucune
 - Sortie : Aucune
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner la tête (module *cercle*) ;
 2. Dessiner le corps (module *triangle*) ;
 3. Dessiner les jambes (module *ligne intersectrices*).

4.4 Module lignes intersectrices

- Analyse
 - Entrée : Aucune
 - Sortie : Lignes intersectrices
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner lignes intersectrices.

4.5 Module rectangle

- Analyse
 - Entrée : Aucune
 - Sortie : Aucune
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner le sommet (module *ligne*);
 2. Dessiner les côtés du rectangle (module *lignes parallèle*);
 3. Dessiner la base (module *ligne*).

4.6 Module cercle

- Analyse
 - Entrée : Aucune
 - Sortie : Dessin d'un cercle
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner cercle.

4.7 Module triangle

- Analyse
 - Entrée : Aucune
 - Sortie : Aucune
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner lignes intersectrices (module *ligne intersectrices*).
 2. Dessiner base (module *ligne*);

4.8 Module ligne

- Analyse
 - Entrée : Aucune
 - Sortie : Dessin d'une ligne horizontale
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner une ligne.

4.9 Module lignes parallèles

- Analyse
 - Entrée : Aucune
 - Sortie : Dessin de deux lignes parallèles
 - Constante : Aucune
 - Formule : Aucune
- Conception
 1. Dessiner deux lignes parallèles.

5 Implantation

```

/** *****/
/*                                                    */
/*  \file   : dessin.cc                               */
/*                                                    */
/*  \brief  Ce programme dessine une maison et la silhouette */
/*          d'une fillette                               */
/*                                                    */
/*  <b>Entrees:</b>                                       */
/*  <b>Sorties:</b>                                       */
/*                                                    */
/*  \author Gabriel Girard                             */
/*  \date: 12 septembre 1994                            */
/*  \version 28 juillet 2003 (G. Girard)                 */
/*  \version 10 septembre 2015 (G. Girard)               */
/*                                                    */
/***** */
#include <iostream>

using namespace std;

int main()
{
    // declaration de fonctions utiles
    void dessine_maison();
    void dessine_fillette();

    // On dessine la maison
    dessine_maison();

    // On dessine la fillette
    dessine_fillette();

} // fin main()

```

```
/** *****/
/*
/*  dessine_fillette
/*
/*  \brief Cette fonction dessine une fillette
/*
/*
/*  \param[in]
/*  \return
/** *****/

void dessine_fillette()
{
    // declaration des fonctions utiles
    void dessine_cercle();
    void dessine_intersect();
    void dessine_triangle();

    // On dessine la fillette
    dessine_cercle(); // la tete
    dessine_triangle(); // le corps
    dessine_intersect(); // les jambes
}

/** *****/
/*
/*  dessine_rectangle
/*
/*  \brief Cette fonction dessine un rectangle
/*
/*
/*  \param[in]
/*  \return
/** *****/

dessine_rectangle()
{
    // declarations des fonctions utiles
    void dessine_base();
    void dessine_parallele();

    // On dessine le rectangle
    dessine_base();
    dessine_parallele();
    dessine_base();
}
```

```
/** *****/
/*
/*  dessine_triangle
/*
/*  \brief Cette fonction dessine un triangle
/*
/*
/*  \param[in]
/*  \return
/** *****/
dessine_triangle()
{
    // declarations des fonctions utiles
    void dessine_base();
    void dessine_intersect();

    // On dessine le rectangle
    dessine_intersect();
    dessine_base();
}
/** *****/
/*
/*  dessine_base
/*
/*  \brief Cette fonction dessine une ligne
/*
/*
/*  \param[in]
/*  \return
/** *****/
dessine_base()
{
    cout << "-----" << endl;
}
/** *****/
/*
/*  dessine_intersect
/*
/*  \brief Cette fonction dessine deux lignes intersectrices
/*
/*
/*  \param[in]
/*  \return
/** *****/
dessine_intersect()
{
    cout <<"    / \\ " << endl;
    cout <<"  /    \\ " << endl;
    cout <<"/      \\ " << endl;
}

```

```
/** *****/
/*
/*  dessine_parallele
/*
/*  \brief Cette fonction dessine deux lignes verticales
/*          paralleles
/*
/*  \param[in]
/*  \return
/*****/
dessine_parallele()
{
    cout <<"|          |" << endl;
    cout <<"|          |" << endl;
    cout <<"|          |" << endl;
}

/** *****/
/*  dessine_cercle
/*
/*  \brief Cette fonction dessine un cercle
/*
/*  \param[in]
/*  \return
/*****/
dessine_cercle()
{
    cout << "   * *"<< endl;
    cout << " *   *"<< endl;
    cout << "   * *"<< endl;
}

```