

Métriques Génie Logiciel

Pierre Gérard
pierre.gerard@iutv.univ-paris13.fr

Licence Pro. SIL
IUT de Villetaneuse - Université de Paris 13

1 Dessiner un graphe de contrôle

```
cin >> a >> b >> c;           // A
if (a>10) {                   // B
    cout << "hello";         // C
    if (b<a) {                // D
        cout << "partie 1";  // E
        if (c>a)             // F
            cout << "partie 2"; // G
    }
    else
        cout << "partie 3";  // H
}
cout << "fin";
```

Question : Dessinez le graphe de contrôle du programme ci-dessus.

2 Mesurer la complexité structurelle d'un programme

```
int main() {
    int a, b, c;               // A
    int petit;                 // B
    cin >> a >> b >> c ;       // C
    if (a==b)                  // D
        return -1;            // E
    if (b==c)                  // F
        return -1;            // G
    if (a==c)                  // H
        return -1;            // I
    if (a>b) {                 // J
        if (b>c)               // K
            petit = 3;         // L
        else
            petit = 2;         // M
    }
    else
        if (a>c)               // N
            petit = 3;         // O
        else
            petit = 1;         // P
    return petit;              // Q
}
```

Question : Calculez de trois façons le nombre cyclomatique de McCabe associé au code ci-dessus.

3 Mesurer la complexité empirique d'un programme

Considérons le pseudo-code suivant :

```
read x,y,z
type="scalène";
if (x==y or x==z or y==z) type="isocèle";
if (x==y and x==z) type="équilatéral";
if (x>y+z or y>x+z or z>x+z) type="pas un triangle"
if (x<=0 or y<=0 or z<=0) type="données erronées"
print type
```

Question : Calculez les métriques de Halstead pour ce programme.

4 Mesurer les spécificités objet d'un programme

```
class personne{
    char* nom;
    char* nss;
public:
    personne(){nom=new char[LONGUEURNOM]; nss=new char[LONGUEURSS];}
    ~personne(){delete nom; delete nss;}
    void ajouteNom(char* nouvNom){strcpy(nom, nouvNom);}
    void ajouteNSS(char* nouvNSS){strcpy(nss, nouvNSS);}
    char* lireNom(){return nom;}
    void virtuel affiche(){cout << "le nom de la personne est " << nom;}
};

class edudiant public personne{
    float note;
public:
    void ajouteNote(float nouvNote){note=nouvNote;}
    void affiche(){cout << "le nom de l'étudiant est " << lireNom() << " et sa note est " << note;}
};

class listePersonne{
    personne* liste[MAX];
    int indexListe;
public:
    listePersonne(){indexListe=0;}
    void ajoutPersonne(char* nouvNom, char* nouvNSS){
        liste[indexListe]=new personne;
        liste[indexListe]->ajoutNom(nouvNom);
        liste[indexListe]->ajoutNSS(nouvNSS);
        indexListe++;
    }
    void ajoutEtudiant(char* nouvNom, char* nouvNSS, float note){
        etudiant* temp = new etudiant;
        temp->ajoutNom(nouvNom); temp->ajoutNSS(nouvNSS);
        temp->ajoutNote(nouvNote);
        liste[indexListe++]=temp;
    }
    void affiche(){
        int j;
        for(j=0; j<indexListe; j++)
            liste[j]->affiche();
    };
};
```

Question : Calculez les métriques de Chidamber pour le code ci-dessus, qui stocke des informations sur des étudiants dans un tableau.

5 Mesurer la cohésion fonctionnelle

```
cin >> a >> b;
int x, y, z;
x = 0; y = 1; z = 1;
while (a>0) {
    x = x + b;
    z = z * b;
    if (a > b)
        y = y * a;
    a = a - 1;
}
cout << x << a << z << y;
```

Question : Calculer le cohésion de ce code en utilisant les métriques de Bieman et Ott. Pour ce faire, dessinez un graphe orienté indiquant les flux.