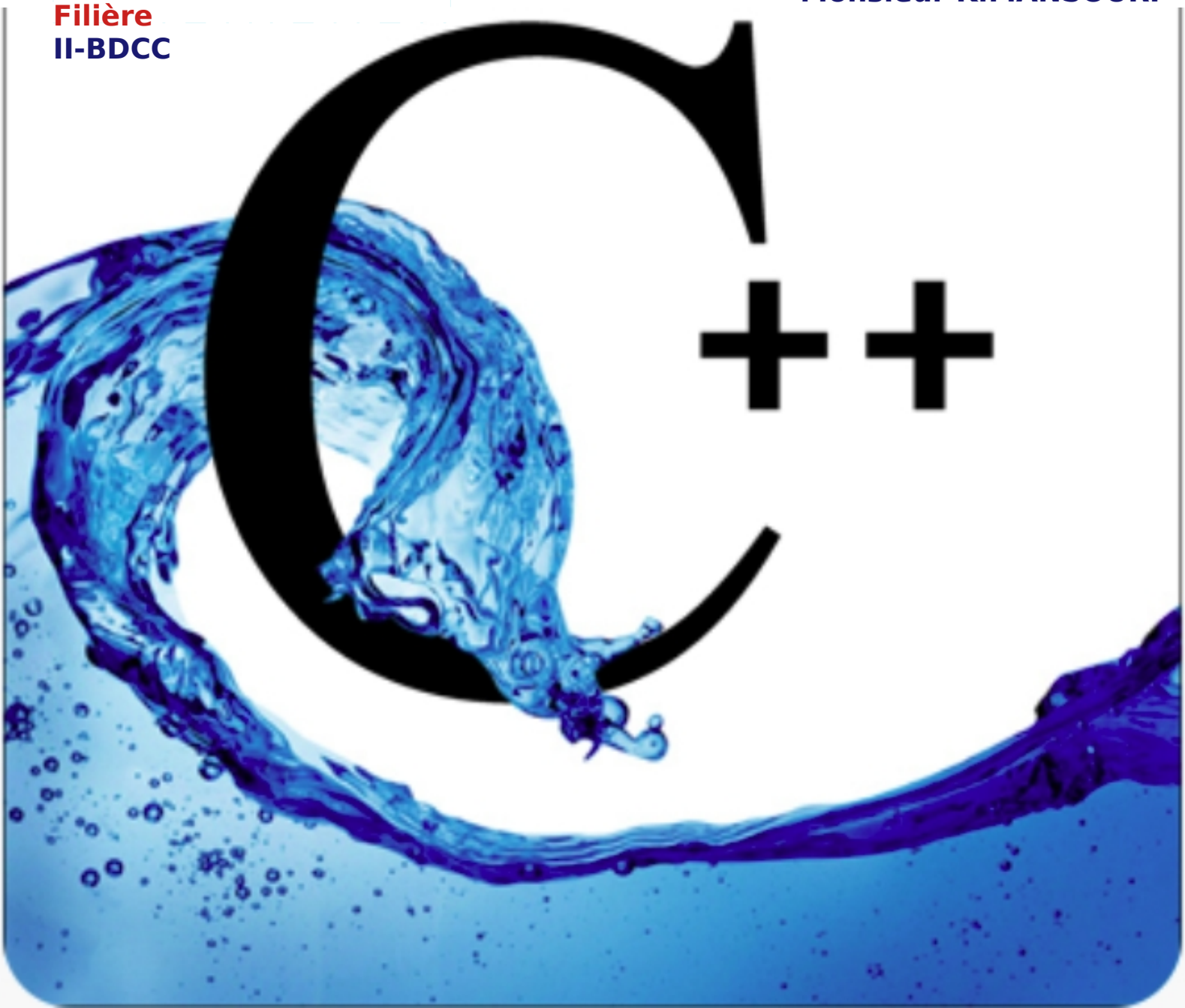
COMPTE RENDU DE CHAPITRE 4 + TPS

Réalisé par :
Abderrahmane ETTOUNANI
Filière
II-BDCC

Encadré par :
Monsieur K.MANSOURI



les exemples de cours & tps

Exemple 01.cpp > ...

```cpp
#include <iostream>
using namespace std;
class point
{
    int x,y;
public:
    point(int,int);
    ~point();
};
point :: point(int abs,int ord)
{
    x=abs;y=ord;
    cout <<"Construction du point " <<x<<"  "<<y<<"\n";
}
point::~point()
{
    cout<<"destruction du point "<<x<<" "<<y<<"\n";
}
void test()
{
    cout <<"Debut de test() \n";
    point u(3,7);
    cout<<"Fin de test()\n";
}
int main()
{
    cout<<"Debut de test()\n";
    point a(1,4);
    test();
    point b(5,10);
    for(int i=0;i<3;i++)
    {
        cout <<"Boucle tour numero " <<i<<"\n";
        point(7+i,12+i);
    }
    cout<<"Fin de Main() \n";
    return 0;
}
```

```
Debut de test()
Construction du point 1  4
Debut de test()
Construction du point 3  7
Fin de test()
destruction du point 3 7
Construction du point 5  10
Boucle tour numero 0
Construction du point 7  12
destruction du point 7 12
Boucle tour numero 1
Construction du point 8  13
destruction du point 8 13
Boucle tour numero 2
Construction du point 9  14
destruction du point 9 14
Fin de Main()
destruction du point 5 10
destruction du point 1 4
```

```cpp
#include <iostream>
using namespace std;
class point
{
    int x,y;
public:
    point(int,int);
    ~point();
};
point :: point(int abs,int ord)
{
    x=abs;y=ord;
    cout <<"Construction du point " <<x<<"  "<<y<<"\n";
}
point::~point()
{
    cout<<"destruction du point "<<x<<" "<<y<<"\n";
}
point a(1,4);
int main()
{
    cout<<"Debut de main()\n";
    point b(5,10);
    cout<<"Fin de Main() \n";
    return 0;
}
```

```
Construction du point 1  4
Debut de main()
Construction du point 5  10
Fin de Main()
destruction du point 5 10
destruction du point 1 4
```

```cpp
#include <iostream>
using namespace std;
class point
{
    int x,y;
public:
    point(int,int);
    ~point();
};
point :: point(int abs,int ord)
{
    x=abs;y=ord;
    cout <<"Construction du point " <<x<<"  "<<y<<"a l'adresse "<<this<<"\n";
}
point::~point()
{
    cout<<"destruction du point "<<x<<" "<<y<<"a l'adresse : "<<this <<"\n";
}
int main()
{
    cout<<"Debut de main()\n";
    point a(0,0);
    a=point(1,2);
    a=point(3,5);
    cout<<"Fin de Main() \n";
    return 0;
}
```

```
Debut de main()
Construction du point 0  0a l'adresse 0x7ffedf0ee140
Construction du point 1  2a l'adresse 0x7ffedf0ee148
destruction du point 1 2a l'adresse : 0x7ffedf0ee148
Construction du point 3  5a l'adresse 0x7ffedf0ee150
destruction du point 3 5a l'adresse : 0x7ffedf0ee150
Fin de Main()
destruction du point 3 5a l'adresse : 0x7ffedf0ee140
```

```cpp
#include <iostream>
using namespace std;
class point
{
public:
    int x,y;
    point(int,int);
    ~point();
};
point :: point(int abs,int ord)
{
    x=abs;y=ord;
    cout <<"Construction du point " <<x<<"  "<<y<<"\n";
}
point::~point()
{
    cout<<"destruction du point "<<x<<" "<<y<<"\n";
}
int main()
{
    void fct(point *);
    point *adr;
    cout<<"Debut de main()\n";
    adr=new point(3,7);
    fct(adr);
    cout<<adr->x<<" . "<<adr->y;
    delete adr;
    cout<<"Fin de Main() \n";
    return 0;
}
void fct(point *adp)
{
    cout <<"Debut de la fonction \n";
    adp->x=4;
    adp->y=6;
    cout<<adp->x<<adp->y<<"Fin de la fonction \n";
}
```

```
Debut de main()
Construction du point 3  7
Debut de la fonction
46Fin de la fonction
4 . 6destruction du point 4 6
Fin de Main()
```

```cpp
#include <iostream>
using namespace std;
class point
{

public:
    int x,y;
    point(int,int);
    ~point();
};
point :: point(int abs,int ord)
{
    x=abs;y=ord;
    cout <<"Construction du point " <<x<<"  "<<y<<"\n";
    cout<<"son adresse est:"<<this<<"\n";
}
point::~point()
{
    cout<<"destruction du point "<<x<<" "<<y<<"\n";
    cout<<"son adresse est:"<<this<<"\n";
}
int main()
{
    cout<<"Debut de main()\n";
    point a(3,7);
    point b=a;
    cout<<"Fin de Main() \n";
    return 0;
}
```

```
Debut de main()
Construction du point 3  7
son adresse est:0x7ffe421aeab8
Fin de Main()
destruction du point 3 7
son adresse est:0x7ffe421aeac0
destruction du point 3 7
son adresse est:0x7ffe421aeab8
```

```cpp
#include <iostream>
using namespace std;
class liste
{
int taille;
float *adr;
public:
    liste(int);
    ~liste();
};
liste :: liste(int t)
{
    taille=t;
    adr=new float[taille];
    cout <<"Construction " << "adresse de l'objet: "<<this<<"\n";
    cout <<"Adresse de la liste : "<<adr<<"\n";}
liste::~liste()
{
    cout<<"destruction de l'objet avec l'adresse \n"<<this<<"\n";
    cout<<"l'adresse de la liste est:"<<adr<<"\n";
    delete adr;
}
int main()
{
    cout<<"Debut de main()\n";
    liste a(3);
    liste b=a;
    cout<<"Fin de Main() \n";
    return 0;
}
```

```
Debut de main()
Construction adresse de l'objet: 0x7fff04760e60
Adresse de la liste : 0x555a3fae8ec0
Fin de Main()
destruction de l'objet avec l'adresse
0x7fff04760e70
l'adresse de la liste est:0x555a3fae8ec0
destruction de l'objet avec l'adresse
0x7fff04760e60
l'adresse de la liste est:0x555a3fae8ec0
```

```cpp
#include <iostream>
using namespace std;
class liste
{
int taille;
float *adr;
public:
    liste(int);
    liste(liste &);
    ~liste();
};
liste::liste(liste &v)
{
    taille=v.taille;
    adr=new float[taille];
    for(int i=0;i<taille;i++)
    {
        adr[i]=v.adr[i];
    }
    cout<<" \n Constructeur par recopie";
    cout<<" Adresse de l'objet : "<<this;
    cout<<" Adresse de la liste : "<<adr<<"\n";
}
liste :: liste(int t)
{
    taille=t;
    adr=new float[taille];
    cout <<"Construction " << "adresse de l'objet: "<<this<<"\n";
    cout <<"Adresse de la liste : "<<adr<<"\n";}
liste::~liste()
{
    cout<<"destruction de l'objet avec l'adresse \n"<<this<<"\n";
    cout<<"l'adresse de la liste est:"<<adr<<"\n";
    delete adr;
}
int main()
{
    cout<<"Debut de main()\n";
    liste a(3);
    liste b=a;
    cout<<"Fin de Main() \n";
    return 0;
}
```

```
Debut de main()
Construction adresse de l'objet: 0x7ffec356cac0
Adresse de la liste : 0x560fa6051ec0

 Constructeur par recopie Adresse de l'objet : 0x7ffec356cad0 Adresse de la liste : 0x560fa6051ee0
Fin de Main()
destruction de l'objet avec l'adresse
0x7ffec356cad0
l'adresse de la liste est:0x560fa6051ee0
destruction de l'objet avec l'adresse
0x7ffec356cac0
l'adresse de la liste est:0x560fa6051ec0
```

```cpp
class point
{
int x,y;
public:
    point(int,int);
    point(point &);
    ~point();
    point symetrique();
    void affiche(){cout<<"x="<<x<<" y="<<y<<"\n";};
};
point::point(point &pt)
{
    x=pt.x;y=pt.y;
    cout<<" \n Constructeur par recopie";
    cout<<" construction du point : "<<x<<" "<<y;
    cout<<" Adresse de l'objet : "<<this<<"\n";
}
point :: point(int abs=0,int ord=0)
{
    x=abs;y=ord;
    cout<<" construction du point : "<<x<<" "<<y;
    cout<<" Adresse de l'objet : "<<this<<"\n";
}
point ::~point()
{
    cout<<"destruction du point "<<x<<" "<<y;
    cout<<"son adresse est "<<this<<"\n";
}
point point::symetrique()
{
    point res;
    cout <<"*****************************************\n";
    res.x=-x;
    res.y=-y;
    cout<<"#########################################\n";
    return res;
}
int main()
{
    cout<<"Debut de main()\n";
    point a(1,4),b;
    cout<<"Avant Appel a Symetrique\n";
    b=a.symetrique();
    b.affiche();
    cout<<"Apres appel a symetrique et Fin de Main() \n";
    return 0;
}
```

```
Debut de main()
 construction du point : 1 4 Adresse de l'objet : 0x7ffd41fa60e0
 construction du point : 0 0 Adresse de l'objet : 0x7ffd41fa60e8
Avant Appel a Symetrique
 construction du point : 0 0 Adresse de l'objet : 0x7ffd41fa60f0
*****************************************
#########################################
destruction du point -1 -4son adresse est 0x7ffd41fa60f0
x=-1 y=-4
Apres appel a symetrique et Fin de Main()
destruction du point -1 -4son adresse est 0x7ffd41fa60e8
destruction du point 1 4son adresse est 0x7ffd41fa60e0
```

```cpp
#include <iostream>
using namespace std;
class liste
{
int taille;
float *adr;
public:
    liste(int);
    liste(liste &);
    void saisie();
    void affiche();
    liste oppose();
    ~liste();
};
liste::liste(liste &v)
{
    taille=v.taille;
    adr=new float[taille];
    for(int i=0;i<taille;i++)
    {
        adr[i]=v.adr[i];
    }
    cout<<" \n Constructeur par recopie";
    cout<<" Adresse de l'objet : "<<this;
    cout<<" Adresse de la liste : "<<adr<<"\n";
}
liste :: liste(int t)
{
    taille=t;
    adr=new float[taille];
    cout <<"Construction " << "adresse de l'objet: "<<this<<"\n";
    cout <<"Adresse de la liste : "<<adr<<"\n";}
liste::~liste()
{
    cout<<"destruction de l'objet avec l'adresse \n"<<this<<"\n";
    cout<<"l'adresse de la liste est:"<<adr<<"\n";
    delete adr;
}
void liste::saisie()
{
    int i;
    for(i=0;i<taille;i++)
    {
        cout<<"Entrer un Nombre : ";
        cin>>*(adr+i);
    }
}
void liste::affiche()
{
    int i;
    for(i=0;i<taille;i++)
    {
        cout<<*(adr+i)<<" ";
    }
    cout<<"\n Adresse de l'objet : "<<this<<" Adresse de liste : "<<adr<<"\n";
}

liste liste::oppose()
{
    liste res(taille);
    for(int i=0;i<taille;i++)
    {
        res.adr[i]=-adr[i];
    }
    for(int i=0;i<taille;i++)
    {
        cout<<res.adr[i]<<" ";
    }
    cout <<"\n";
    return res;
}

int main()
{
    cout<<"Debut de main()\n";
    liste a(3),b(3);
    a.saisie();
    a.affiche();
    b=a.oppose();
    b.affiche();
    cout<<"Fin de Main() \n";
    return 0;
}
```

```
Debut de main()
Construction adresse de l'objet: 0x7ffd17d16290
Adresse de la liste : 0x555f09e9b2c0
Construction adresse de l'objet: 0x7ffd17d162a0
Adresse de la liste : 0x555f09e9b2e0
Entrer un Nombre : 2
Entrer un Nombre : 4
Entrer un Nombre : 5
2 4 5
 Adresse de l'objet : 0x7ffd17d16290 Adresse de liste : 0x555f09e9b2c0
Construction adresse de l'objet: 0x7ffd17d162b0
Adresse de la liste : 0x555f09e9b760
-2 -4 -5
destruction de l'objet avec l'adresse
0x7ffd17d162b0
l'adresse de la liste est:0x555f09e9b760
0 0 5.59874e-33
 Adresse de l'objet : 0x7ffd17d162a0 Adresse de liste : 0x555f09e9b760
Fin de Main()
destruction de l'objet avec l'adresse
0x7ffd17d162a0
l'adresse de la liste est:0x555f09e9b760
```

```cpp
#include <iostream>
using namespace std;

class point
{
    int x,y;
public :
    point(int abs=0,int ord=0)
    {
        x=abs;
        y=ord;
        cout<<"Constructeur point "<<x<<" "<<y<<"\n";
    }
};

class poincol
{
    point p;
    int couleur;
public:
    poincol(int,int,int);
};

poincol :: poincol(int abs,int ord,int coul) :p(abs,ord)
{
    couleur =coul;
    cout <<"Constructeur poincol "<<couleur<<"\n";
}

int main()
{
    poincol a(1,3,9);
    return 0;
}
```

```
Constructeur point 1 3
Constructeur poincol 9
```

----------partie 1 :----------

```cpp
class note
{
    float value;
public :
    void input();
    void print();
    void set (float);
    float get();
    char* apprecier();
};
void note :: input ()
{
    cout << "saisisez la note : ";
    cin >> value;
    if (value > 20 || value <0)
    {
        cout << "la note ne doit pas etre >20 ou <0: ";
        note :: input ();
    }
}
void note :: print ()
{
    cout << "la note est : " << value;
}
void note :: set(float v)
{
    value = v;
}
float note :: get ()
{
    return value;
}
char* note :: apprecier ()
{
    if (value >= 16)
    return "tres bien";
    else if (value <= 15 && value >=12)
    return "bien";
    else if (value <12 && value >=10)
    return "passable";
    else return "mauvaise note";
}
main()
{
    note a;
    a.input();
    char* chaine = a.apprecier();
```

```
(base) ettounani@tounani
saisisez la note : 10

passable
(base) ettounani@tounani
saisisez la note : 8

mauvaise note
(base) ettounani@tounani
saisisez la note : 18

tres bien
(base) ettounani@tounani
```

----------partie 2----------

rtie 02.cpp > harmonise(note &)

```cpp
#include <iostream>
using namespace std;
class note
{
    float value;
public :
    note (float v)
    {
        value = v;
    }
    void input();
    void print();
    void set (float);
    float get();
    char* apprecier();
    void harmonise(note&);
    void moyenne(note *, int);
    char* apprecier(note*, int);
};
void note :: input (){
    cout << "saisissez la note : ";
    cin >> value;
    if (value > 20 || value<0)
    {
        cout << "la  note ne doit pas etre >20 ou <0 : ";
        note :: input ();
    }
}
void note :: print (){
    cout << "la note est : " << value;
}
void note :: set(float v){
    value = v;
}
float note :: get (){
    return value;
}
char* note :: apprecier (){
    if (value >= 16)
    return "tres bien";
    else if (value <= 15 && value >=12)
    return "bien";
    else if (value <12 && value >=10)
    return "passable";
    else return "mauvaise note";
}
```

```
void harmonise(note &a){
    if (a.get() < 8)
    a.set(0);
    else a.set(8);
}
float moyenne(note *t, int n)
{
    float somme=0;
    for (int i = 0; i < n; i++)
    somme += t[i].get();
    return (somme/n);
}
void apprecier(note *t, int n)
{
    for (int i = 0; i < n; i++)
    {
    cout <<"\n" << t[i].get() <<" .. "<< t[i].apprecier() <<"\n";
    }
}
main()
{
    note T[4] = {13, 10, 4, 12, 19};
    cout <<"\n" << moyenne(T, 4)<<"\n";
    apprecier(T, 4);

}
```

```
9.75

13 .. bien

10 .. passable

4 .. mauvaise note

12 .. bien
```

----------partie 3----------

```cpp
#include <iostream>
using namespace std;
class note
{
    float value;

public :

    note (float v)
    {
        value = v;
    }
    void input();
    void print();
    void set (float);
    float get();
    char* apprecier();
    void harmonise(note&);
    void moyenne(note *, int);
    char* apprecier(note*, int);


};

void note :: input ()
{
    cout << "saisissez la note : ";
    cin >> value;
    if (value > 20 || value<0)
    {
        cout << "la  note ne doit pas etre >20 ou <0 : ";
        note :: input ();
    }
}

void note :: print ()
{
    cout << "la note est : " << value;
}

void note :: set(float v)
{
    value = v;
}

```

```cpp
float note :: get ()
{
    return value;
}

char* note :: apprecier ()
{
    if (value >= 16)
    return "tres bien";
    else if (value <= 15 && value >=12)
    return "bien";
    else if (value <12 && value >=10)
    return "passable";
    else return "mauvaise note";
}

void harmonise(note &a)
{
    if (a.get() < 8)
    a.set(0);
    else a.set(8);
}


float moyenne(note *t, int n)
{
    float somme=0;
    for (int i = 0; i < n; i++)
    somme += t[i].get();
    return (somme/n);
}


void apprecier(note *t, int n)
{
    for (int i = 0; i < n; i++)
    {
    cout <<"\n" << t[i].get() <<" .. "<< t[i].apprecier() <<"\n";
    }
}
```

```cpp
int main()
{

    int n;
    float v;
    note *T;
    cout << "  donnez le nbr d'eleves : ";
    cin >> n;
    T = new note[n];

    for(int i = 0; i < n; i++)
    {
        cout << "\n Donner note : ";
        cin >> v;
        T[i].set(v);
    }
    apprecier(T, n);
    cout <<"\n moyenne des notes avant Harmonise " << moyenne(T, n) <<"\n";

    for (int i = 0; i < n; i++)
    {
        if (T[i].get() < 15)
        harmonise(T[i]);
    }
    cout <<"\n moyenne des notes apres Harmonise " << moyenne(T, n) <<"\n";
}
```

```
  donnez le nbr d'eleves : 4

 Donner note : 13

 Donner note : 5

 Donner note : 15

 Donner note : 20

13 .. bien

5 .. mauvaise note

15 .. bien

20 .. tres bien

 moyenne des notes avant Harmonise 13.25

 moyenne des notes apres Harmonise 10.75
```