

## T.P. 4

# Initiation au langage C++

## Les librairies

```
#include <d:\sinclair\sources\cpp\robot.h>
#include <stdio.h>

void main()
{
    robot r1;
    robot r2;
    r1.avance (5); r1.tourne ();   r1.avance (2);
    r2.tourne ();   r2.setorientation ('O');r2.avance (3); r2.tourne ();
    printf ("Robot 1 : X = %d Y = %d \n",r1.Xposition (),r1.Yposition ());
    printf ("Robot 2 : X = %d Y = %d \n",r2.Xposition (),r2.Yposition ());
}
```

## L'héritage : robot.h

```
class rob
{
    int Xpos;
    int Ypos;
    char Orientation;

public :
    rob ();
    int Xposition (void);
    int Yposition (void);
    void setorientation (char);
    void avance(int n);
    void tourne(void);
};

int rob::Xposition (void)
    { return (Xpos); }

int rob::Yposition (void)
    { return (Ypos); }

void rob::setorientation (char c)
    { Orientation=c; }

void rob::avance(int n)
{
    switch (Orientation)    {
        case 'N': {Ypos=Ypos+n; break;}
        case 'E': {Xpos=Xpos+n; break;}
        case 'S': {Ypos=Ypos-n; break;}
    }
```

```

        case 'O': {Xpos=Xpos-n; break;}
    } }

void rob::tourne(void)
{
    switch (Orientation)    {
        case 'N': {Orientation='E'; break;}
        case 'E': {Orientation='S'; break;}
        case 'S': {Orientation='O'; break;}
        case 'O': {Orientation='N'; break;}
    } }

rob::rob ()
{
    Xpos=0;   Ypos=0;
    Orientation='N'; }

class rhp:public rob
{
    int pinceau;
public:
    void av (char);
    void crayon(void);
    int crayonstate(void);
    int Xrhp(void);
    int Yrhp(void);
    rhp(void);
};

void rhp::av(char c)
{
    setorientation (c);
    avance(1);
}

void rhp::crayon()
{
    if (pinceau==1)
        pinceau=0;
    else
        pinceau=1;
}

int rhp::crayonstate()
{
    return (pinceau);
}

int rhp::Xrhp()
{
    return (Xposition()+200);
}

```

```

}

int rhp::Yrhp()
{
    return (200-Yposition()+200);
}

rhp::rhp()
{
    rob();
    pinceau=1;
}

```

## La tortue : Robot.h

```

#ifndef _ROBOT_H
#define _ROBOT_H

class rob
{
    int Xpos;
    int Ypos;
    char Orientation;

public :
    rob ();
    int Xposition (void);
    int Yposition (void);
    void setorientation (char);
    void avance(int n);
    void tourne(void);
};
#endif

```

## La tortue : Robot.cpp

```

#include <stdafx.h>
#include "d:\sinclair\sources\cpp\robot.h"

rob::rob ()
{
    Xpos=0;    Ypos=0;
    Orientation='N';
}

int rob::Xposition (void)
    {    return (Xpos); }

int rob::Yposition (void)

```

```

        {      return (Ypos); }

void rob::setorientation (char c)
    {      Orientation=c; }

void rob::avance(int n)
{      switch (Orientation)  {
        case 'N': {Ypos=Ypos+n; break;}
        case 'E': {Xpos=Xpos+n; break;}
        case 'S': {Ypos=Ypos-n; break;}
        case 'O': {Xpos=Xpos-n; break;}
        }      }

void rob::tourne(void)
{      switch (Orientation)  {
        case 'N': {Orientation='E'; break;}
        case 'E': {Orientation='S'; break;}
        case 'S': {Orientation='O'; break;}
        case 'O': {Orientation='N'; break;}
        }      }

```

## La tortue : Rhp.h

```

#ifndef _RHP_H
#define _RHP_H
#include "d:\sinclair\sources\cpp\robot.h"

class rhp:public rob
{
    int pinceau;
public:
    void av (char);
    void crayon(void);
    int crayonstate(void);
    int Xrhp(void);
    int Yrhp(void);
    rhp(void);
};

#endif

```

## La tortue : Rhp.cpp

```

#include <stdafx.h>
#include "d:\sinclair\sources\cpp\rhp.h"

void rhp::av(char c)
{
    setorientation (c);
}

```

```

        avance(1);
    }

void rhp::crayon()
{
    if (pinceau==1)
        pinceau=0;
    else
        pinceau=1;
}

int rhp::crayonstate()
{
    return (pinceau);
}

int rhp::Xrhp()
{
    return (Xposition()+200);
}

int rhp::Yrhp()
{
    return (200-Yposition());
}

rhp::rhp()
{
    rob();
    pinceau=1;
}

```

## La tortue : tortueDlg.cpp

```

void CTortueDlg::OnButtonTrace()
{
    ttue.crayon();
    // TODO: Add your control notification handler code here
}

void CTortueDlg::OnButtonNord()
{
    ttue.av ('N');

    if (ttue.crayonstate()==1)
    {
        CClientDC dc(this) ;
        int x=ttue.Xrhp();
        int y=ttue.Yrhp();
    }
}

```

```

        dc.SetPixel(x,y,RGB(0,0,0)) ;
    }

}

void CTortueDlg::OnButtonOuest()
{
    ttue.av ('O');

    if (ttue.crayonstate()==1)
    {
        CClientDC dc(this) ;
        int x=ttue.Xrhp();
        int y=ttue.Yrhp();
        dc.SetPixel(x,y,RGB(0,0,0)) ;
    }

}

void CTortueDlg::OnButtonSud()
{
    ttue.av ('S');

    if (ttue.crayonstate()==1)
    {
        CClientDC dc(this) ;
        int x=ttue.Xrhp();
        int y=ttue.Yrhp();
        dc.SetPixel(x,y,RGB(0,0,0)) ;
    }

}

void CTortueDlg::OnButtonEst()
{
    ttue.av ('E');

    if (ttue.crayonstate()==1)
    {
        CClientDC dc(this) ;
        int x=ttue.Xrhp();
        int y=ttue.Yrhp();
        dc.SetPixel(x,y,RGB(0,0,0)) ;
    }

}

```