FND 17

Point2D.h

```
typedef struct point2d
{
    double x;
    double y;
} Point2D;
Point2D *mallocPoint2D();
void freePoint2D(Point2D *pThis);
Point2D *createPoint2D(double x, double y);
void setPoint2D(Point2D *pPt, double x, double y);
Point2D *copyPoint2D(Point2D *pThis);
double getXPoint2D(Point2D *pThis);
double getYPoint2D(Point2D *pThis);
double getDistancePoint2D(Point2D *pThis, Point2D *pThat);

```

Point2D.c

```c
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "Point2D.h"

Point2D *mallocPoint2D()
{
    Point2D *pPt;
    pPt = (Point2D *)malloc(sizeof(Point2D));
    if (pPt == (Point2D *)NULL)
    {
        return NULL;
    }
    else
    {
        return pPt;
    }
}

void freePoint2D(Point2D *pThis)
{
    free(pThis);
}

Point2D *createPoint2D(double x, double y)
{
    Point2D *pt;
    pt = mallocPoint2D();
    pt->x = x;
    pt->y = y;
    return pt;
}
```

```
33
34  void setPoint2D(Point2D *pPt, double x, double y)
35  {
36      pPt->x = x;
37      pPt->y = y;
38  }
39
40  Point2D *copyPoint2D(Point2D *pThis)
41  {
42      Point2D *copy = mallocPoint2D();
43      copy->x = pThis->x;
44      copy->y = pThis->y;
45      return copy;
46  }
47
48  double getXPoint2D(Point2D *pThis)
49  {
50      return pThis->x;
51  }
52
53  double getYPoint2D(Point2D *pThis)
54  {
55      return pThis->y;
56  }
57
58  double getDistancePoint2D(Point2D *pThis, Point2D *pThat)
59  {
60      double xd = pow((pThis->x - pThat->x), 2);
61      double yd = pow((pThis->y - pThat->y), 2);
62      return sqrt(xd + yd);
63  }
```

Discussion

This concept is helpful in programming because it allows us as programmers to create our own data types which will simplify programming. In this example (FND) it allows us to create points (coordinates) data types. These points have x and y coordinates. So, whenever we create a point, we expect it to have x and y coordinates. And in this FND we can perform the operations specified. Another example is what we did in FND 16 when we used typedef char* String. It allowed us to use String instead of char* which helps with simplicity and familiarity.