Lab 5.3.10 Points in 2D: part 3

Objectives

Familiarize the student with:

- modelling real-world entities with classes and objects;
- interactions between objects of the same type.

Scenario

Use the code from the previous exercise as a starting point.

It's sometimes useful in computer graphics to detect if three or more points in a 2D space are collinear.

Use the code from your previous example and add the method contains() to the Line2D class that will check if a given point belongs to that line.

Your program should read in two sets of x and y coordinates and construct an object of the class Line2D based on those two points.

Then, your program should read a third point and determine if it's a part of the same line.

The output of your program should just be the verdict "collinear" or "not collinear".

Remember that comparing floating-point values in C++ might be a bit tricky.

```cpp
#include <iostream>

using namespace std;

class Point2D{
public:
  Point2D(double x, double y);
  string toString();
  double getX();
  double getY();
  // ...
private:
  double x;
  double y;
};

class Line2D{
public:
  Line2D(double slope, double y_intercept);
  Line2D(Point2D pointA, Point2D pointB);
  string toString();
  bool contains(Point2D point);
  // ...
private:
  double slope;
  double y_intercept;
};

// implement Point2D and Line2D methods
```

Example input

```
0 0
3 6
-2 -4
```

Example output

```
collinear
```

Example input

```
2 2
-2 0
0 0
```

Example output

```
not collinear
```