Object-oriented programming

Practice 3

Hyung-Sin Kim

Seoul National University
Graduate School of Data Science

3.1 Cartesian Plane

- Write two classes *Point* and *Line* that give the same output as below.

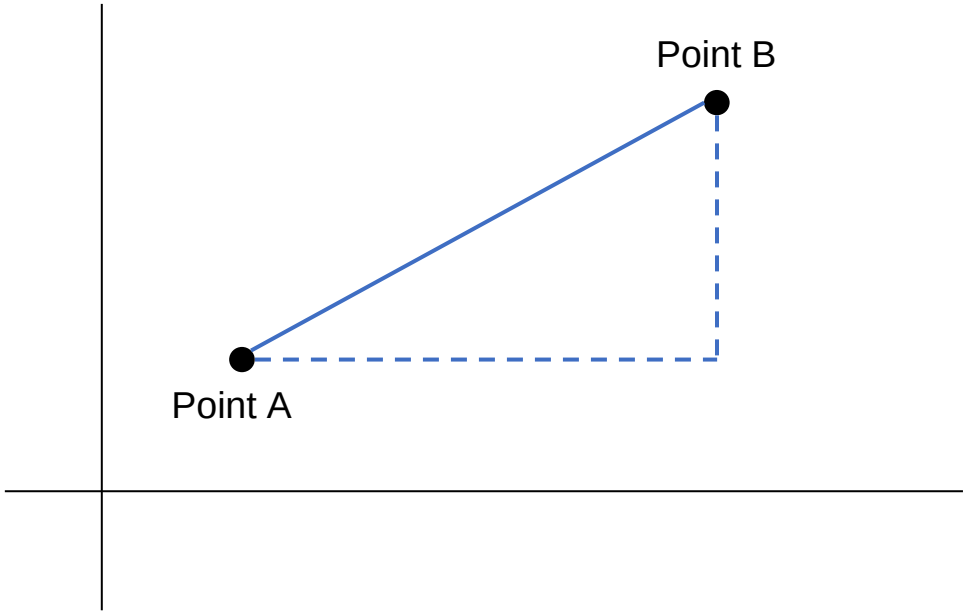
```
line = Line(Point(1,1), Point(3,2))
```

```
line.slope()
```

```
0.5
```

```
line.length()
```

```
2.23606797749979
```



3.2-1 Write class Country

- By using **__init__** method, make class *Country* that takes name, population, area as parameters

```
canada = Country("Canada", 34482779, 9984670)
print(canada.name)
print(canada.area)
```

```
Canada
9984670
```

- Define **is_larger** method.
  is_larger : a method that returns 'True' only when the first country has a larger area than the second country. (if not, return 'False')

```
usa = Country("United State of America", 313914040, 9826675)
canada.is_larger(usa)
```

```
True
```

- Define **population_density** method.
  population_density : a method that returns the population density (number of people per area)

```
canada.population_density()
```

```
3.4535722262227995
```

3.2-2 Write class Continent

- By using **__init__** method, make class Continent that takes name and countries as parameters
  - Countries : a list of Country objects
  - Countries has more than one element

```
canada = Country('Canada', 34482779, 9984670)
usa = Country('United States of America', 313914040, 9826675)
mexico = Country('Mexico', 112336538, 1943950)
countries = [canada, usa, mexico]
north_america = Continent('North America', countries)
north_america.name
```

```
'North America'
```

- Define **total_population** method
  total_population : a method that returns the sum of the population of countries belonging to the continent

```
north_america.total_population()
```

```
460733357
```

Thanks!

hyungkim@snu.ac.kr