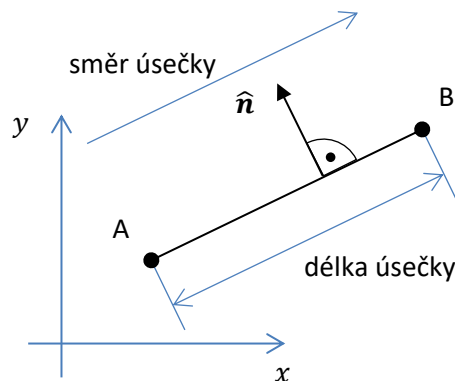


Cílem šesté úlohy je naimplementovat modul pro práci s úsečkami, který následně využijeme v sedmém zadání. Omezíme se na úsečky ležící v rovině, přičemž každá úsečka bude určena dvěma krajními body. Rovněž definujeme specializovanou třídu reprezentující 2D vektor. Všechny uvedené třídy objektů vybavíme základními operacemi.



Následující body by vás měly postupně provést detaily implementace jednotlivých úkolů. Při návrhu struktur se přidrže deklarací uvedených v šabloně.

Úkol 1: (1 bod)

Vytvoření třídy Point.

Navrhněte strukturu nové třídy Point, která bude reprezentovat bod ve 2D rovině. Vedle konstruktoru doplňte třídu o metodu pro výpočet rozdílu dvou bodů (přetížením operátoru -) a pro výpočet nového bodu posunutého o zadaný translační vektor (přetížením operátoru +). Uvedené vektory jsou definovány v následujícím úkolu. Umožněte rovněž přetypování objektu na řetězec ve formátu "[x, y]".

```
def main():
    a = Point(2, 3)
    print(a) # out: [2.000, 3.000]
```

Úkol 2: (1 bod)

Vytvoření třídy Vector.

Navrhněte strukturu nové třídy Vector, která bude reprezentovat vektor ve 2D rovině. Tato třída bude potomkem třídy Point. Rozšiřte třídu Vector o metodu length vracející délku vektoru (L2 norma). Dále přidejte metodu normalize, která provede normalizaci vektoru, tj. z vektoru nenulové délky udělá vektor jednotkové délky. Přetížením operátoru * umožněte provést násobení vektoru skalárem, výsledkem bude nový vektor. Doplňte metodu dot pro výpočet skalárního součinu (ang. dot product) dvou vektorů [1]. Stejně jako v případě bodu umožněte přetypování vektoru na řetězec ve formátu

"(x, y)". Popřemýšlejte nad přetížením operátorů - a + pro rozdíl a součet dvou vektorů (obě operace dávají v případě vektorů smysl).

```
def main():
    a = Point(2, 3)
    print(a) # out: [2.000, 3.000]
    b = Point(9, 4)
    d = b - a
    print(d) # out: (7.000, 1.000)
    print(d * 3) # out: (21.000, 3.000)
    print(d.length()) # out: 7.0710678118654755
    d.normalize()
    print(d.length()) # out: 1.0
```

Úkol 3: (1 bod)

Vytvoření třídy Line.

Naimplementujte třídu Line, která bude reprezentovat úsečku zadanou dvěma body. Třidu doplňte o metodu `direction` vracející rozdíl dvou krajních bodů úsečky. Dále vytvořte metodu `normal`, která vypočte normálový vektor dané úsečky (normálový vektor úsečky je vektor kolmý na směrový vektor úsečky přičemž orientace této normály by měla odpovídat obrázku z první strany). Stejně jako v předešlých případech umožněte přetypování úsečky na řetězec ve formátu "[A.x, A.y] -> [B.x, B.y]".

```
def main():
    ...
    line = Line(a, b)
    print(line) # out: [[2.000, 3.000] -> [9.000, 4.000]]
    normal = line.normal()
    print(normal) # out: (-1.000, 7.000)
    print(normal.dot(d)) # out 0.0
```

Reference

[1] https://cs.wikipedia.org/wiki/Skal%C3%A1rn%C3%AD_sou%C4%8Din