Programiranje 1 — šesta domača naloga

Rok za oddajo: petek, 5. decembra 2025

Točka in premica

Naloga

Napišite razreda `Tocka` in `Premica` v skladu s sledečimi navodili.

Razred `Tocka`

Razred `Tocka` napišite tako, da bodo njegovi objekti predstavljali posamezne točke v ravnini. Točka je določena s koordinatama x in y. V razredu realizirajte sledeče konstruktorje in metode:[1]

- `public Tocka(double x, double y)` [1–50]

  Inicializira objekt, ki predstavlja točko s koordinatama x in y.

- `public double vrniX()` [1–50]

  Vrne koordinato x točke `this` (torej točke, ki jo predstavlja objekt, na katerega kaže kazalec `this`, če smo natančnejši).

- `public double vrniY()` [1–50]

  Vrne koordinato y točke `this`.

- `public String toString()` [1]

  Vrne niz oblike

  `(`x`,␣`y`)`

  pri čemer x in y podajata koordinati točke `this`, zaokroženi na dve decimalki. Na primer:

  `(-2.65,␣15.30)`

- `public double razdalja(Tocka t)` [2–5]

  Vrne evklidsko razdaljo med točkama `this` in `t`. Evklidska razdalja med točkama (x_1, y_1) in (x_2, y_2) znaša $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

- `public static Tocka izhodisce()` [6]

  Vrne objekt, ki predstavlja koordinatno izhodišče, torej točko $(0, 0)$.

- `public double razdaljaOdIzhodisca()` [7–10]

  Vrne evklidsko razdaljo točke `this` od koordinatnega izhodišča.

[1] Navajamo samo javno dostopne elemente razreda; privatnih ne predpisujemo. V oglatih oklepajih so navedene številke skritih testnih primerov, v katerih se lahko pripadajoči element pokliče.

Razred `Premica`

Razred `Premica` napišite tako, da bodo njegovi objekti predstavljali posamezne premice v ravnini. Ukvarjali se bomo samo s premicami, ki niso niti vodoravne niti navpične. Takšna premica je določena z enačbo $y = kx + n$, kjer je k ($k \neq 0$) smerni koeficient, n pa začetna vrednost. V razredu realizirajte sledeče javno dostopne konstruktorje in metode:

- `public Premica(double k, double n)` [11–50]

  Izdela objekt, ki predstavlja premico z enačbo $y = \texttt{k}x + \texttt{n}$.

- `public double vrniK()` [11–50]

  Vrne smerni koeficient premice `this`.

- `public double vrniN()` [11–50]

  Vrne začetno vrednost premice `this`.

- `public String toString()` [11]

  Vrne niz oblike

  `y`␣`=`␣k␣`x`␣`+`␣n

  pri čemer k in n podajata smerni koeficient in začetno vrednost premice `this`, zaokrožena na dve decimalki. Na primer:

  `y = -7.00 x + -21.52`

- `public Tocka tockaPriX(double x)` [12–15]

  Vrne točko s podano koordinato x, ki leži na premici `this`.

- `public static Premica skoziTocko(double k, Tocka t)` [16–20]

  Vrne premico s smernim koeficientom `k`, ki potuje skozi točko `t`.

- `public Premica vzporednica(Tocka t)` [21–24]

  Vrne vzporednico premice `this`, ki potuje skozi točko `t`.

- `public Premica pravokotnica(Tocka t)` [25–28]

  Vrne pravokotnico na premico `this`, ki potuje skozi točko `t`. (Če ima premica smerni koeficient k, je smerni koeficient njene pravokotnice enak $-1/k$.)

- `public Tocka presecisce(Premica p, double epsilon)` [29–33]

  Če sta premici `this` in `p` »približno« vzporedni (če se njuna smerna koeficienta razlikujeta za manj kot `epsilon`), vrne `null`, sicer pa vrne presečišče premic `this` in `p`.

- `public Tocka projekcija(Tocka t)` [34–37]

  Vrne pravokotno projekcijo točke `t` na premico `this`. Pomagajte si s sliko 1.

- `public double razdalja(Tocka t)` [38–41]

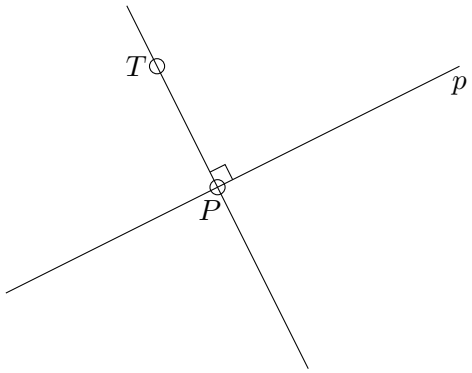
  Vrne pravokotno razdaljo med premico `this` in tocko `t`. Pomagajte si s sliko 1.

- `public double razdaljaOdIzhodisca()` [42–45]

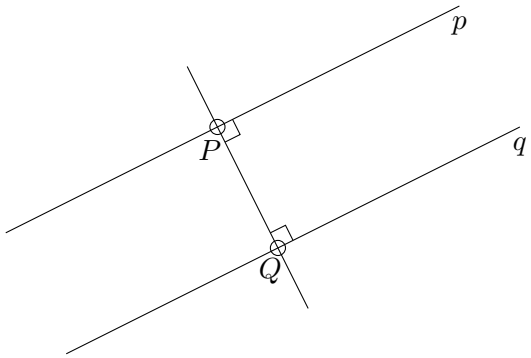
  Vrne pravokotno razdaljo med premico `this` in koordinatnim izhodiščem.

- `public double razdalja(double n)` [46–50]

Vrne pravokotno razdaljo med premico `this` in premico, ki ima enak smerni koeficient kot premica `this`, njena začetna vrednost pa znaša `n`. Pomagajte si s sliko 2.



Slika 1: *Točka P je pravokotna projekcija točke T na premico p. Pravokotna razdalja med točko T in premico p je definirana kot dolžina daljice TP.*



Slika 2: *Dolžina daljice PQ je pravokotna razdalja med premicama p in q.*

Izpis decimalnega ločila

Če v javi izpišemo realno število ali pa ga pretvorimo oz. vključimo v niz, se lahko decimalno ločilo prikaže kot pika ali vejica. Izbira je v splošnem skladna s sistemskimi jezikovnimi nastavitvami. Pri tej nalogi pa zahtevamo, da se decimalno ločilo prikaže kot pika. To lahko (neodvisno od sistemskih jezikovnih nastavitev) dosežemo s posebno različico metode `String.format`:

`String.format(java.util.Locale.US,` *format*`,` *arg1*`, ...,` *argN*`)`

Pri tem so *format* in *arg1*, ..., *argN* parametri osnovne različice metode `String.format` (tj. formatni niz in vrednosti, s katerimi se nadomestijo podnizi oblike `%x` v formatnem nizu).

Oddaja naloge

POZOR! Pri izdelavi razredov `Tocka` in `Premica` izhajajte iz datotek `Tocka.java` in `Premica.java`, ki ju najdete na Učilnici v pripadajočem arhivu zip. **Imen datotek ne spreminjajte**; ostaneta naj `Tocka.java` in `Premica.java`.

Oddajte obe datoteki, tudi če katere od njih morebiti ne boste spreminjali. Datotek ne »zipajte«, ampak vsako posebej oddajte na Učilnico.