

1. V Mathematici lahko točko v ravnini predstavimo s parom dveh koordinat (npr. $\{1, 2\}$). Trikotnik lahko podamo s seznamom treh točk A , B in C , navedenih v smeri nasprotni smeri urinega kazalca. Kot primer vzami trikotnik podan v obliki:

```
trikotnik = {{0, 0}, {5, 1}, {7, 4}}
```

Sestavi naslednje funkcije:

- `Stranice[{AA_, BB_, CC_}]`, ki vrne seznam stranic a , b in c trikotnika, pri čemer $\{AA_, BB_, CC_$ predstavlja trikotnik. Stranice so daljice, ki jih podamo kot par (seznam) dveh točk. Pozor: podvojena imena točk uporabljamo, ker je spremenljivka (ime) C v Mathematici že zasedena.
- `Koti[{AA_, BB_, CC_}]`, ki vrne seznam kotov α , β in γ trikotnika, pri čemer kot podamo kot zaporedje treh točk. Npr. kot α določajo točke C , A , B v tem zaporedju. Točke naj bodo vedno navedene v smeri nasprotni urinem kazalcu (t.i. pozitivna orientacija).
- `SlikaOglisc[trikotnik_]`, ki vrne seznam točk kot grafičnih objektov (`Point`), ki ga lahko uporabimo za izris, če ga umestimo v objekt `Graphics`. Uporabi funkcijo `Map`. Točke naj bodo nekoliko večje.
- `SlikaStranic[trikotnik_]`, ki vrne seznam povezav kot grafičnih objektov (`Line`). Črte naj bodo sive barve.
- `NarisiTrikotnik[trikotnik_]`, ki nariše trikotnik z oglišči in stranicami. Uporabi zgornji funkciji. Izris naj bo v pravem razmerju (`AspectRatio`).

2. Sestavi funkcijo `VektorSimetralkota[{x_, y_, z_}]`, ki za dano trojico točk, ki predstavlja kot trikotnika, izračuna enostski vektor v smeri simetralkota. Sestavi še funkcijo `Simetralkota[{x_, y_, z_}, dol_]`, ki vrne daljico, ki predstavlja simetralo kota in se začne v točki y , je usmerjena v notranjost kota in je dolžine `dol`. Argument `dol` naj bo opcijski in ima privzeto vrednost 10. Namig: par $\{a, b\}$ lahko normaliziraš z ukazom `Normalize`.

3. S pomočjo funkcij v prejšnji nalogi sestavi še `SlikaSimetralkotov[trikotnik_]`, ki vrne grafično predstavitev treh daljic (dolžine 10), katere predstavljajo `NarisiTrikotnik[trikotnik_]` tako, da bo pri uporabi opcije `Simetralkota -> True` izrisala še simetralko (privzeta vrednost opcije naj bo `False`; uporabi funkcijo `OptionsPattern` ter `OptionValue`).

4. Sestavi funkcijo `PresecisceSimetral[{AA_, BB_, CC_}]`, ki za dan trikotnik $\{AA_, BB_, CC_$ vrne presečišče simetral kotov (t.j. točko). Ker vemo, da se vse tri simetralko kotov sekajo v isti točki, zadostuje, da poiščemo presečišče dveh simetral. Pomagaj si s funkcijo `Module`.

5. Sestavi funkcijo `NajblizjaTockaNaPremici[{AA_, BB_}, XX_]`, ki za dano točko XX poišče najbližjo točko na premici, ki gre skozi točki AA in BB . Pomagaj si s funkcijo `Module`.

6. Sestavi še funkcijo `RadijVcrtanegaKroga[{AA_, BB_, CC_}]`. Ta je definiran kot razdalja med presečiščem simetral kotov ter poljubno stranico (oz. premico nosilko stranice). Pomagaj si s funkcijama iz prejšnjih dveh nalog in s funkcijo `Module`.

7. Nadgradi funkcijo `NarisiTrikotnik[trikotnik_]` še z opcijo `VcrtanKrog`, s privzeto vrednostjo `VcrtanKrog -> False`, ki krmili, ali narišemo še očrtan krog. Ta naj dodatno nariše še presečišče

8. S pomočjo funkcije `Manipulate[...]` in funkcije `NarisiTrikotnik[trikotnik_]` sestavi interaktivno animacijo, kjer točko $CC = cx, cy$ trikotnika spreminjaš lahko tako, da velja $cx \in [-7, 7]$ ter $cy \in [3, 6]$.