Kryžiukai-Nuliukai (Tic Tac Toe)

1. Įvadas

Kas yra jūsų programa?

Tai žaidimas „Kryžiukai-Nuliukai“ (Tic Tac Toe), parašyta Python kalba taikant objektinio programavimo (OOP) principus. Naudotojas gali žaisti prieš dirbtinį intelektą (AI), o žaidimų rezultatai yra saugomi CSV faile.

Kaip paleisti programą?

Norint paleisti programą, reikia turėti įdiegtą Python. Vykdykite: python tic\_tac\_toe.py

Kaip naudotis programa?

Žaidėjai paeiliui deda simbolius („X“ arba „O“) į 3x3 dydžio lentą. Laimi tas, kuris pirmas sudėlioja tris simbolius iš eilės. Jei visi langeliai užpildyti ir nėra laimėtojo – lygiosios.

2. Analizė

4 Objektinio programavimo principai

- Inkapsuliacija

Visi duomenys (pvz., lentos būsena) inkapsuliuoti klasėse Board, Player ir Game. Prieiga prie jų vykdoma per metodus.

- Abstrakcija

Player klasė yra abstrakti bazinė klasė (abc modulis). Ji apibrėžia bendrą sąsają tiek žmogaus, tiek AI žaidėjams.

- Paveldėjimas

HumanPlayer ir AIPlayer klasės paveldi Player klasę ir perrašo get\_move() metodą.

- Polimorfizmas

get\_move() metodas veikia polimorfiškai – kiekvienas žaidėjas (žmogus arba AI) pateikia savo versiją, kuri naudojama vienodai.

Dizaino šablonas: Gamyklos metodas (Factory Method)

- Naudotas šablonas: Factory Method

Klasė PlayerFactory naudoja statinį metodą create\_player(), kuris sukuria žmogų arba AI žaidėją – atskiriant kūrimo logiką nuo žaidimo logikos.

- Kodėl šis šablonas tinkamas?

Jis leidžia lankščiai ir lengvai pridėti naujų žaidėjų tipų ateityje (pvz., tinklinį žaidėją).

Kompozicija ir/ar agregacija

- Klasė Game agreguoja Board ir Player objektus. Jie nėra tvirtai susieti – galima juos keisti nepriklausomai, rodo agregacijos principų laikymąsi.

Skaitymas ir rašymas į failą

- Programa įrašo kiekvieno žaidimo rezultatą į game\_results.csv.

- Ji taip pat skaito ir rodo ankstesnių žaidimų rezultatus – taip įgyvendinamas duomenų išsaugojimas ir atvaizdavimas.

Testavimas

- Įtraukta vienetinių testų (unittest) patikrinanti:

* Lentos atnaujinimą
* Laimėtojo nustatymą
* Lygiosiomis pasibaigusio žaidimo atvejį
* Galimų ėjimų grąžinimą

3. Rezultatai

* Programa atitinka visus keturis objektinio programavimo principus.
* Pritaikytas Factory Method dizaino šablonas žaidėjų kūrimui.
* Rezultatai įrašomi ir skaitomi iš CSV failo.
* Pagrindinės funkcijos padengtos vienetiniais testais.
* Programa pritaikyta naudoti aplinkose be input(), naudojant iš anksto nustatytus ėjimus.

4. Išvados

Šis projektas parodo, kaip galima taikyti objektinio programavimo koncepcijas ir dizaino šablonus kuriant paprastą žaidimą Python kalba. Jame aiškiai integruoti abstrakcijos, inkapsuliacijos, paveldėjimo ir polimorfizmo principai. Pasirinktas dizaino šablonas (Factory Method) ir duomenų išsaugojimas faile padidina programos lankstumą ir funkcionalumą. Ateityje būtų galima išplėsti projektą pridedant grafinę naudotojo sąsają (GUI), tinklinį daugelio žaidėjų režimą ar pažangesnį AI.