BALTIC TALENTS ACADEMY

JAVA ABSTRAKČIOS KLASĖS

TESTAS #8 - KLASĖS, LAUKAI, METODAI

- Priskyrimas pagal reikšmę
- Parametrų perdavimas pagal reikšmę
- Priskyrimas pagal nuorodą
- Parametrų perdavimas pagal nuorodą
- Objekto laukų pradinės reikšmės pagal nutylėjimą
- Klasės laukai
- Klasės metodai

ABSTRAKTI KLASĖ

- Abstrakti klasė tai tokia klasė:
  - Jos kai kurie metodai nėra iki galo aprašyti, t.y. apie juos žinome tik jų pasiekiamumo tipą, pavadinimą, parametrus ir kokią reikšmę jie grąžina
  - Mes nenorime (!!!), kad galima būtų sukurti tokios klasės objektą ir norime (!!!), kad ji būtų naudojama tik kaip tėvinė klasė kitoms klasėms aprašyti
- Kad klasė yra abstrakti nurodoma su modifikatoriumi abstract

ABSTRAKTI KLASĖ

- Abstrakti klasė gali turėti ne iki galo aprašytus metodus, t.y. tokius metodus kuriems nurodyta tik jų pasiekiamumo tipas, pavadinimas, parametrai ir kokią reikšmę jie grąžina

```
public abstract class Figure {
    private String color;
    public Figure(String color) {
        this.color = color;
    }
    public abstract double getArea();
}
```

Demo 1

POLIMORFIZMAS (POLYMORPHISM)

- Polimorfizmas – objektiniame programavime naudojama sąvoka, kai operacija (metodas) gali būti vykdomas skirtingai, priklausomai nuo konkrečios klasės (ar duomenų tipo) realizacijos, metodo kvietėjui nieko nežinant apie tokius skirtumus ir galvojant, kad jis operuota tik su bazinės (tėvinės) klasės objektais
- Tai pasiekiama aprašant metodus (operacijas) bazinėje klasėje ir perrašant atitinkamus metodus paveldinčiose klasėse. Metodai, kuriuos galima (paliekant tą patį pavadinimą) perrašyti paveldinčioje klasėje, vadinami virtualiais.

- *iš wikipedia*

POLIMORFIZMAS (POLYMORPHISM)

- Kai dukterinėje klasėje perrašomas tėvinės klasės metodas, tai galima tą metodą dukterinėje klasėje pažymėti specialia žyme (anotacija) @Override
- Taip mes pažymime, kad žinome, jog šio metodo aprašas (vardas, grąžinamos reikšmės tipas, parametrai) sutampa su tėvinės klasės metodu
- Tyčia ar netyčia pakeitus tėvinio ar dukterinio metodo aprašą ir atsiradus neatitikimams, mums bus rodomas klaidos pranešimas

Demo 2

SUBTIPAS / SUPERTIPAS

class Mokinys extends Zmogus { … }

- Mes žinom, kad Mokinys tai **dukterinė** klasė klasės Zmogus atžvilgiu
- Savo ruožtu Zmogus yra **tėvinė** klasė klasei Mokinys.
- Ji taip pat vadinama **bazine (base)** klase klasei Mokinys
- Klasė Mokinys yra **subtipas (subtype)** klasei Zmogus
- Klasė Zmogus yra **supertipas (supertype)** klasei Mokinys

VIENAS IŠ PAGRINDINIŲ OOP PRINCIPŲ (ABSTRAKCIJOS)

- Kiekvienas subtipo kintamasis tuo pačiu yra supertipo
- T.y. ten kur galima naudoti supertipo klases kintamuosius ar parametrus, ten taip pat galima naudoti ir subtipo.
- Aprašant kintamuosius ar metodų parametrus reikia stengtis naudoti kuo bendresnes klases

```
class Circle extends Figure { … }
```

- Ten kur galima naudoti Figure objektus galima naudoti ir Circle objektus, nes kiekvienas Circle objektas yra ir Figure tipo objektas.

Demo 3

UŽDAVINYS

Sukurkite abstrakčią klasę Figura kurioje būtų aprašyti abstraktūs metodai plotui ir perimetrui paskaičiuoti

Tada sukurkite dukterines klases Apskritimas, Kvadratas, Trikampis (lygiakraštis).

1. Paskaičiuokite kokie turėtų būti visų perimetrai, kad plotai būtų vienodi, tarkime lygūs 100.
2. Paskaičiuokite kokie turėtų būti visų plotai, kad perimetrai būtų vienodi, tarkime lygūs 100.

Pastaba: apsirašykite Figura klasėje tokius abstrakčius metodus ir po to juos aprašykite dukterinėse klasėse, kad kaip parametrą pateikus plotą arba perimetrą, jie paskaičiuotų ir nustatytų atitinkamos figūros kraštinę ar spindulį.