

BALTIC TALENTS ACADEMY

JAVA SAŞSAJA (INTERFACE)

TESTAS #9 - ABSTRAKTI KLASĖ, POLIMORFIZMAS

- ▶ Abstrakti klasė
- ▶ Polimorfizmas (Polymorphism)
- ▶ Subtipas / supertipas

SĄSAJOS (INTERFACE)

- ▶ Vienas būdas realizuoti abstrakcijas yra su abstrakčiomis klasėmis
- ▶ Kitas būdas - tai interfeisai

```
interface Area {  
    void setSizeFromArea(double area);  
    ...  
}
```

- ▶ Pagal nutylėjimą interfeise nurodyti metodai turi **public** prieinamumą
- ▶ Interfeiso aprašas primena abstrakčios klasės aprašymą, kur visi metodai yra abstraktūs

SĄSAJOS (INTERFACE)

- ▶ Kaip naudoti interfeisus aprašant klases?
- ▶ Aprašant mes turime nurodyti, kad mūsų klasė įgyvendina interfeisą, t.y. jame aprašytus metodus

```
class Circle extends Figure implements Area {  
  
}
```

- ▶ Mes privalome klasėje Circle pilnai aprašyti visus metodus nurodytus sąsajoje Area. Prie jų taip pat labai naudinga nurodyti žymę (anotaciją) **@Override**

SĄSAJOS PRAPLĖTIMAS

- ▶ Kaip ir klasės taip ir sąsajos gali išplėsti/papildyti kitas sąsajas papildomais metodais

```
interface Geometry extends Area {  
    void setSizeFromPerimeter(double perimeter);  
    ...  
}
```

KELIŲ SAŠAJŲ ĮGYVENDINIMAS

- ▶ Kai žinote, klasė gali būti dukterine klase tik vienai klasei, t.y. ji gali turėti vieną ir tik vieną tėvinę klasę
- ▶ Bet klasė gali įgyvendinti ne vieną, o kelias sąsajas

```
class Figure implements Area, Color, Border {
```

```
    ...
```

```
}
```

- ▶ Toks užrašas reiškia, kad klasėje **Figure** aprašyti/įgyvendinti visi metodai nurodyti ir **Area** ir **Color** ir **Border** sąsajose.

SĄSAJOS DEFAULT METODAI

8

- ▶ Sąsajoje galima turėti ne tik metodų aprašus (signature), bet ir jų įgyvendinimą (implementation)
- ▶ Tokie metodai sąsajoje pažymimi modifikatoriumi **default**
- ▶ Tokiu būdu klasėje, kuri įgyvendina sąsają, nebūtina (!) tokį metodą įgyvendinti.

SĄSAJOS DEFAULT METODAI



- ▶ Jei klasė įgyvendina dvi sąsajas, kurios abi turi tokį patį **default** metodą (t.y. toks pats pavadinimas ir parametrai), tai klasė privalo įgyvendinti metodą, nes java nežino kurį iš metodų naudoti

SĄSAJOS STATIC METODAI



- ▶ Panašiai kaip sąsajos **default** metodus, galima aprašyti **static** metodus
- ▶ Nuo **default** jie skiriasi tuo, kad pasidaro priskirti ne prie objekto egzemplioriaus (instance), o prie klasės - kaip kad klasės metodai
- ▶ **Static** metodai gali būti kviečiami iš kitų sąsajos metodų, arba kaip sąsajos metodai (kaip kad klasės metodai)

SĄSAJOS PRIVATE METODAI



- ▶ Java9 jau galima apibrėžti privačius (**private**) metodus, kurie veikia kaip ir **default** bet yra prieinami iš kitų sąsajos metodų, bet ne iš klasės ar klasės objekto.
- ▶ Analogiškai kaip **static** metodai galima aprašyti **private static** metodus, kurie taip pat bus prieinami tik iš sąsajos.

SĄSAJOS TIPO KINTAMIEJI / PARAMETRAI

- ▶ Sąsają galima naudoti nurodant kintamųjų ar metodo parametrų tipus
- ▶ Tokio tipo kintamiesiems ar parametrams galima priskirti tokius objektus, kurių klasė įgyvendina duotą sąsają (t.y. kurių klasė implementuoja duotą interfeisą)

UŽDAVINYS

Aprašykite interfeisą Mokėjimas (Payment) su šiais metodais:

- ▶ banko sąskaita (bank account) - gražina banko sąskaitos numerį
- ▶ sąskaitos turėtojas (account owner) - fizinio ar juridinio asmens pavadinimas
- ▶ suma (amount) - pervedama suma

Sukurkite klases Darbuotojas (Employee) ir Klientas (Client), kurios implementuoja Mokėjimas interfeisą. Sukurkite masyvą iš keleto darbuotojų ir klientų objektų ir cikle išveskite jų mokėjimo informaciją