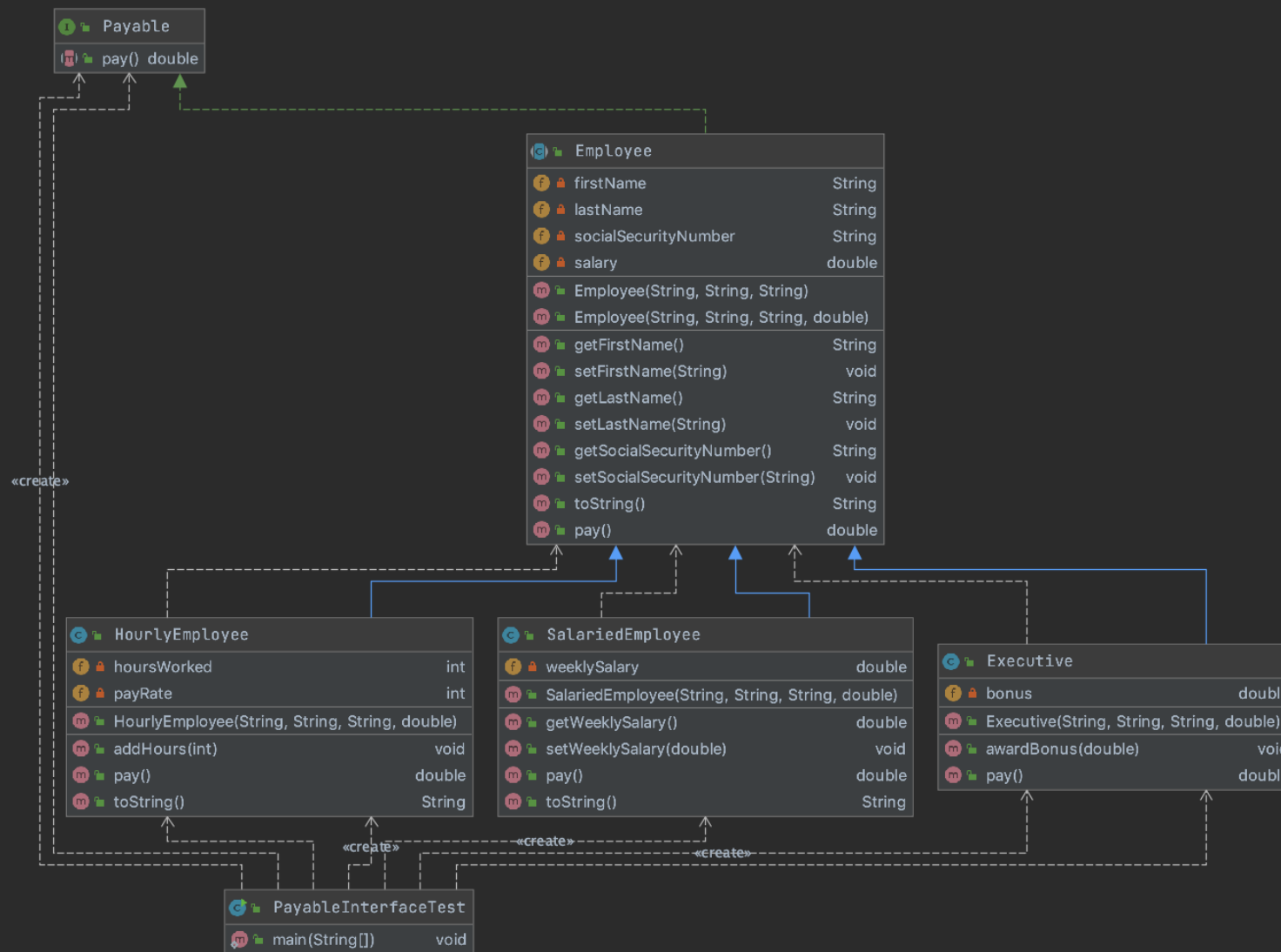


📄 java-8-nodarbiba-klases-1-uzdevums-obligats.md

Tēma 8 - Objektu elementi - Interfeisi, abstraktas klases, mantošana

Uzdevums 1 - Klašu pārmantojamība un polimorfisms

Uzdevuma risināšanai izmantojam sekojošu modeli



- Izveidojam interfeisu Payable un tajā vienu metodi pay
- Izveidojam abstraktu klasi Employee. Šī klase implementē interfeisu Payable. Šī klase ir abstrakta jo nav domāta tiešai izmantošanai bet gan specializētu apakšklasu veidošanai. Šajā klasē ievietosim kopējās loģikas kodu un apakšklasēs tikai specializētu kodu.
 - Klasē ievietojam laukus: firstName,lastName un SSN (Social Security Number) un salary (alga)
 - Izveidojam 2 konstruktorus ar 3 un 4 parametriem, iekļaujam vienā no tiem salary.
 - Ievietojam pay metodi, jo to prasa Payable interfeiss.
 - Noģenerējam setterus/getterus laukiem (izmantojot IntelliJ Generate -> Encapsulation)
- Izveidojam specializētu Employee klasi nosaucot to HourlyEmployee.
 - Šajā klasē ievietojam laukus "int hoursWorked" un "double payRate".
 - aizvietojam (override) metodi pay kura atgriež hoursWorked * payRate. Pēc metodes izpildes hoursWorked = 0.
 - Noģenerējam setterus/getterus laukiem (izmantojot IntelliJ Generate -> Encapsulation)
- Izveidojam specializētu Employee klasi nosaucot to SalariedEmployee.
 - Šajā klasē ievietojam laukus "double weeklySalary".
 - aizvietojam (override) metodi pay kura atgriež weeklySalary.
 - Noģenerējam setterus/getterus laukiem (izmantojot IntelliJ Generate -> Encapsulation)
- Izveidojam specializētu Employee klasi nosaucot to Executive.
 - Šajā klasē ievietojam lauku "double bonus" un metodi awardBonus(double) ar kuru pielikt bonusu.
 - pay metode atgriež virsklases pay metodes rezultātu ar pieskaitītu bonus. Pēc metodes izpildes bonus = 0.

- aizvietojam (override) metodi pay kura atgriež `hoursWorked * payRate`.
- Nogenerējam setterus/getterus laukiem (izmantojot IntelliJ Generate -> Encapsulation)
- Izveidojam `PayInterfaceApp` klasi un tajā sekojošu main metodi

```
// create four-element Payable array
Payable[] payableObjects = new Payable[5];
//
payableObjects[0] =
    new SalariedEmployee("John", "Smith", "111-11-1111", 800.00);
payableObjects[1] =
    new SalariedEmployee("Lisa", "Barnes", "888-88-8888", 1200.00);
//
payableObjects[2] = new HourlyEmployee("Barn", "Nobles", "273-56-288", 15.0);
((HourlyEmployee) payableObjects[2]).addHours(120);
payableObjects[3] = new HourlyEmployee("Scott", "Tiger", "123-13-789", 14.0);
((HourlyEmployee) payableObjects[3]).addHours(144);
//
payableObjects[4] = new Executive("", "", "132-56-654", 1400d);
((Executive) payableObjects[4]).awardBonus(1000);

for (Payable currentPayable : payableObjects) {
    if(currentPayable!=null)
        System.out.println("" +
            currentPayable +
            " payment due = " + currentPayable.pay());
}
```

izpildot klasi parādās aptuveni šāds rezultāts. Tajā redzams darbinieka vārds,uzvārds, ssn, algas komponentes un algas lielums.

```
> Task :PayableInterfaceTest.main()
salaried employee: Employee John Smith 111-11-1111 weekly salary: $800.00 payment due = 800.0
salaried employee: Employee Lisa Barnes 888-88-8888 weekly salary: $1,200.00 payment due = 1200.0
hourly employee: Employee Barn Nobles 273-56-288 rate : $15.00 hoursWorkeed $120 payment due = 1800.0
```

hourly employee: Employee Scott Tiger 123-13-789 rate : \$14.00 hoursWorkeed \$144 payment due = 2016.0
executive employee: Employee 132-56-654 salary plus award = : \$1,000.00 payment due = 2400.0

Uzdevums 2 - create second

Soli