

Zadaci za vježbu iz teme 4 (Nasljeđivanje. Polimorfizam.)

1. Napravite klasu `Person` koja opisuje neku osobu. `Person` sadrži atribute `name` (`String`) `surname` (`String`), `age` (`int`). Napišite konstruktor, `get` i `set` metode, te metode `toString` i `equals` (dvije osobe su jednake ako imaju isto ime i prezime te broj godina).
2. Napravite klase `Student` i `Teacher` koje nasljeđuju klasu `Person`, `Student` sadrži atribut `studentId` (`String`) i `academicYear` (`short int`), a `Teacher` sadrži atribute `email` (`String`), `subject` (`String`) i `salary` (`double`). Napišite konstruktore za sve parametre, `get` i `set` metode, te metode `toString` i `equals` (dva studenta su jednaka ako imaju isti `studentId`, neovisno o ostalim podacima, a dva nastavnika su jednaka ako imaju isti `email`, neovisno o ostalim podacima). Dodatno, u klasi `Teacher` napišite metodu `increaseSalary` koja ne vraća ništa, a prima jedan argument tipa `integer` (koji predstavlja postotak). Metoda treba povećati plaću nastavnika za zadani postotak. Također, napišite i statičku metodu `increaseSalary` koja prima varijabilni broj argumenata, prvi je argument tipa `integer` (koji predstavlja postotak), a ostali su objekti tipa `Teacher` kojima je potrebno povećati plaću za zadani postotak.
3. Napišite glavni program u kojem ćete kreirati polje od 5 osoba i u njega staviti 3 nastavnika i dva studenta. Nakon toga program treba u petlji ispisati ime i prezime svake osobe te na kraju petlje prosječnu plaću svih nastavnika koji se pojavljuju u polju. Također, za sljedeći isječak koda:

```
Person p1 = new Person("Ivo", "Ivic", 20);
Person p2 = new Person("Ivo", "Ivic", 20);
Person p3 = new Student("Ivo", "Ivic", 20, "0036312123", (short)3);
Person p4 = new Student("Marko", "Marić", 25, "0036312123", (short)5);

System.out.println(p1.equals(p2));
System.out.println(p1.equals(p3));
System.out.println(p3.equals(p4));
```

Očekuje se ovakav ispis:

```
true
false
true
```

4. Modeliramo potrebne klase i funkcionalnost za Rent-a-car kuću. Kompanija iznajmljuje vozila iz sljedećih kategorija: automobili, kombi vozila i limuzine. Napravite klasu `Vehicle` koja opisuje neko vozilo. Klasa ima sljedeće privatne attribute: `regNo` (String), `model` (String), `year` (int) i `price` (double, označava cijenu unajmljivanja vozila po satu). Dodajte klasi konstruktor koji prima kao parametre sve vrijednosti za attribute. Napravite `get` i `set` metode za svaki atribut, kao i metodu `toString`.
5. Napravite klase `Car`, `Van` i `Limo` koje nasljeđuju `Vehicle`. Dodatno, `Car` sadrži atribut `carType`(String), `Van` sadrži atribut `height`(double) i `noOfSeats`(int), a `Limo` sadrži attribute `length`(double) te boolean varijable `miniBar` i `sunRoof`. Napravite klase `PassengerVan` i `CargoVan` koje nasljeđuju klasu `Van`, a dodatno imaju privatne attribute `noOfPassengers`(int, za `PassengerVan`) tj. `maxLoad` (double, u kg, za `CargoVan`). Napravite `get` i `set` metode za svaki atribut svih klasa.
6. U svim klasama napišite metodu `getPricePerDay`, i to tako da, za objekte tipa `Vehicle` i `Car` metoda vraća cijenu iz varijable `price` pomnoženu s 24, za putnička kombi vozila vraća 80% cijene po danu pomnožene s 24, za teretna kombi vozila 110% cijene * 24, a za limuzine 150% cijene * 24. Također, u klasi `Vehicle` napišite metodu `getPricePerMonth`, koja će vratiti cijenu vozila ako se unajmljuje na mjesec dana (30 * odgovarajuća cijena po danu). Ovu metodu označite kao `final`. U klasi `Vehicle` napišite i statičku metodu `newestVehicle` koja prima varijabilni broj parametara tipa `Vehicle`, a vraća vozilo koje ima najnoviju godinu proizvodnje (ako ih je više, vrati prvo po redu). Dodajte argument double `cargoSpace` u klasu `Car` (označava volumen prtljažnika). U klasi `Vehicle` napišite statičku metodu `printAllVehiclesWithCargoSpaceGreaterThan` koja kao prvi argument prima double broj koji predstavlja traženu zapreminu, a nakon toga slijedi varijabilni broj argumenata tipa `Vehicle`. Metoda treba ispisati sva vozila koja imaju zapreminu teretnog prostora ili prtljažnika veću od tražene.

Za isječak koda koji slijedi:

```
Vehicle v = new Car("DA1234AA", "Renault Clio", 2001, 20, "coupe", 200);
Car car = new Car("DA8818BB", "Renault Megane Grandtour", 2007, 25, "caravan", 800);
Van van1 = new CargoVan("DA0009PO", "Volkswagen Transporter", 2018, 28, 2, (short)3, 4500);
PassengerVan van2 = new PassengerVan("DA6282EA", "IMV 1600", 1978, 35, 2, (short)3, 0);
Vehicle limo = new Limo("DA2238AB", "Zastava 750 LE", 1983, 220, 3.2, false, false);

System.out.println(v.getModel() + " price per day: " + v.getPricePerDay());
System.out.println(van1.getModel() + " price per day: " + van1.getPricePerDay());
System.out.println(van2.getModel() + " price per month: " + van2.getPricePerMonth());

Vehicle newest = Vehicle.getNewestVehicle(v, car, van1, van2, limo);
System.out.println("Newest: " + newest.getModel() + ", " + newest.getYear());

Vehicle.printAllVehiclesWithCargoSpaceGreaterThan(500, v, car, van1, van2, limo);
```

očekuje se ovakav ispis:

```
Renault Clio price per day: 480.0
Volkswagen Transporter price per day: 739.2
IMV 1600 price per month: 20160.0
Newest: Volkswagen Transporter, 2018
Vehicles with cargo space greater than 500.0 litres:
- Renault Megane Grandtour: 800.0
- Volkswagen Transporter: 4500.0
```