

Zadatak 1

Zadatak: Simulacija sistema zdravstvene ustanove

Karakteristike sistema zdravstvene ustanove su:

- Doktor (ime, prezime, specijalnost) ima više pacijenata (ime, prezime, jmbg, broj zdravstvenog kartona).
- Pacijent može da ima samo jednog izabranog doktora.
- Doktor može da zakaže laboratorijski pregled za pacijenta.
- Svaki laboratorijski pregled ima datum i vreme kada je zakazan
- Tipovi laboratorijskog pregleda su:
 - krvni pritisak (gornja vrednost, donja vrednost, puls)
 - nivo šećera u krvi (vrednost, vreme poslednjeg obroka)
 - nivo holesterola u krvi (vrednost, vreme poslednjeg obroka)

Napraviti simulacionu skriptu koja radi sledeće:

1. kreirati doktora "Milan"
2. kreirati pacijenta "Dragan"
3. pacijent "Dragan" bira doktora "Milan" za svog izabranog lekara
4. doktor "Milan" zakazuje pregled za merenje nivoa šećera u krvi za pacijenta "Dragan"
5. doktor "Milan" zakazuje pregled za merenje krvnog pritiska za pacijenta "Dragan"
6. pacijent "Dragan" obavlja laboratorijski pregled za merenje nivoa šećera u krvi. Simulirati i prikazati rezultate.
7. pacijent "Dragan" obavlja laboratorijski pregled za merenje krvnog pritiska. Simulirati i prikazati rezultate.

Iz sledećeg koda se može zaključiti o klasama koje treba napisati (nisu pomenute sve potrebne klase):

```
$docMilan = new Doktor('Milan', 'Milanovic', 'kardiolog');
$pacDragan = new Pacijent('Dragan', 'Jovanovic', '023342323', '0005677');

$pacDragan->izaberiLekara($docMilan);

$pregled1 = new PregledNivoSecera('12-12-2017', '08:00', $pacDragan);
$docMilan->zakaziPregled($pregled1);
$pregled2 = new PregledKrvniPritisak('12-12-2017', '08:15', $pacDragan);
$docMilan->zakaziPregled($pregled2);

$pregled1->obaviPregled();
$pregled2->obaviPregled();
```

Implementirati klase tako da ovaj kod radi kao što je očekivano.
Ubaciti type hint u metode.
Postaviti apstraktne klase i metode na odgovarajuća mesta.

Zadatak 2

(Nastavak na prethodni zadatak)

Napraviti klasu Logger, sa statickim metodama:

- logujKreiranjeDoktora(Doktor \$doktor)
- logujKreiranjePacijenta(Pacijent \$pacijent)
- logujBiranjeLekara(Pacijent \$pacijent, Doktor \$doktor)
- logujObavljanjePregleda(LaboratorijskiPregled \$pregled)

Dodati logovanje akcija u sistemu (logovanje neka samo radi ispis). Akcije logovati u formatu [datum vreme] [akcija]. Primer jedne linije loga:

[20.03.2013 19:30] Pacijent "Milan" izvrsio je laboratorijski pregled nivoa holesterola u krvi.

Akcije koje treba da se loguju su:

- kreiranje doktora
- kreiranje pacijenta
- pacijent bira doktora
- obavljanje laboratorijskog pregleda

Datum se može dobiti pomoću: (new DateTime())->format('d.m.Y H:i')

Zadatak 3

Simulacija proizvodne i prodajne mreže elektronskih uređaja.

Karakteristike sistema su:

1. kompanija poseduje razlicite sektore
 - a. fabrike
 - b. prodajna mesta
 - c. odsek za nabavke
 - d. odsek za marketing
2. moguće je postaviti pravila oblačenja u odseku za marketing i prodajnim mestima
3. moguće je zakazivati sastanke u odseku za nabavke i odseku za marketing
4. moguće je definisati striktno radno vreme u fabrikama i prodajnim mestima

Napraviti odgovarajuće klase sistema. Postaviti interfejs za tačke 2. 3. i 4.

Zadatak 4

Doraditi program iz prethodnog zadatka sa sledećim karakteristikama:

- svaki sektor ima više zaposlenih
- fabrike proizvode monitore, tastature i laptopove
- svako prodajno mesto i svaka fabrika imaju na lageru nekoliko proizvoda
- kupac ima mogućnost da kupi proizvod u prodajnom mestu

Identifikovati i postaviti klase. Iskoristiti apstraktne klase gde je potrebno.

Napraviti simulacionu skriptu koja radi sledeće:

- Kreirati kompaniju "BenComputers"
- Dodati u kompaniju 4 fabrike, 6 prodajnih mesta, 1 odsek za nabavke i 2 odseka za marketing
- U odsecima za marketing postaviti pravilo oblačenja na "poslovno odelo, kravata nije obavezna"
- Zakazati sastanak u odseku za nabavke za 20. 10. 2017., 8:00 na ime "Mile Teslic"
- Definirati radno vreme u fabrikama od 8h do 16h
- Za svako prodajno mesto i fabriku kreirati proizvoljan broj proizvoda na lageru
- Za svako prodajno mesto dodati po 2 zaposlena
- Kreirati kupca i simulirati kupovinu laptopa i monitora na jednom prodajnom mestu

Zadatak 5

Kreirati klasu Soba koja predstavlja hotelsku sobu. Bitni podaci za hotelsku sobu su

- broj sobe,
- broj kreveta,
- cena,
- da li je soba zauzeta ili ne.

Kreirati klasu Hotel koja omogućava dodavanje soba u hotelu.

U klasi Hotel definisati metod pronadji(\$n) koja proverava da li u hotelu postoji soba u koju se može smestiti \$n osoba, pa ako postoji, kao rezultat daje broj najjeftinije sobe u koju taj broj osoba može da se smesti. Ako ima više takvih soba, rezultat treba da bude broj sobe koja ima najmanji broj kreveta, a ako i takvih ima više, onda od njih vratiti najmanji broj sobe. Ukoliko takva soba ne postoji, rezultat je -1. Napisati i metod zarada(\$n) koji izračunava kolika je ukupna zarada od zauzetih soba sa \$n kreveta. Ako je \$n === 0, onda se računaju sve sobe, bez obzira na broj kreveta.

Zadatak 6

Napisati apstraktnu klasu Instrument i u njoj obezbediti:

- konstruktor bez argumenata
- metod ispisi koji ispisuje sve osobine instrumenta
- apstraktne metode:
 - imeInstrumenta,
 - tipInstrumenta,
 - sviraj,
 - nastimujSe,
 - imaZice,
 - imaDugmice,
 - imaUdaraljke,
 - osnovaOdMetala,
 - osnovaOdPlastike,
 - osnovaOdDrveta.

Napisati klase ZicanInstrument, DuvackiInstrument, UdarackiInstrument (i dalje apstraktne klase, mogu da predefinisu neke metode, npr tipInstrumenta, imaZice, imaDugmice, imaUdaraljke).

Napisati klase:

- Violina,
- Viola,
- Violoncelo,
- Harfa,
- Truba,
- Tuba,
- Horna,
- Flauta,
- Kontrabas,
- Bubanj,
- Ksilofon,
- Saksofon,
- Trombon,
- Timpan,

vodeći računa o tome od koje klase nasleđuju.

Napisati funkciju osobine koja za proizvoljan instrument ispisuje sve osobine instrumenta.

Napisati program koji formira niz Orkestar koji se sastoji od različitih instrumenata.

Ispisati osobine svih instrumenata, zatim naštimovati sve instrumente i na kraju pozvati sve

instrumente da sviraju.

U klasi Instrument obezbediti statičku promenljivu \$brojac koja će brojati koliko instrumenata trenutno ima u opticaju. Obezbediti ispravno povećavanje vrednosti ovog brojača prilikom formiranja instrumenata.

U svakoj klasi obezbediti konstruktor bez argumenata. Radi ilustracije redosleda pozivanja konstruktora pozvati ove metode.