1. Kreirati klasu **Boja** koja ima talasnu dužinu u nm, zasićenje i intenzitet.  
   Napraviti konstruktor kome se prosleđuju tri vrednosti koje treba postaviti.  
   Napraviti odgovarajuće getere i setere.  
   Napraviti funkciju ***ispitajSličnost*** koja za dve prosleđene boje vraća true ukoliko su te dve boje iste, u suprotnom vraća false.

Napraviti funkciju ***srednjaTalasnaDuzina*** koja za niz boja vraća vrednost srednje talasne dužine.

1. Kreirati klasu Datum koja sadrži tri atributa - *dan, mesec i godinu*.  
   Napraviti konstruktor kome se prosleđuju tri vrednosti koje treba postaviti.  
   Napraviti odgovarajuće getere i setere.  
   Napraviti funkciju ***ranijiDatum*** koja za dva prosleđena datuma ispisuje raniji od ta dva datuma. Ukoliko su datumi jednaki onda na ekranu ispisati “Datumi su jednaki”.
2. Kreirati klasu Radnik koja sadrži dva atributa - ime i platu ranika.  
   Napraviti konstruktor kome se prosleđuju dve vrednosti koje treba postaviti.  
   Napraviti odgovarajuće getere i setere.

Napraviti funkciju ***maksPlata*** kojoj se prosleđuje niz radnika, a ona vraća najveću platu ranika.

Napraviti funkciju ***maksPlataOsoba*** kojoj se prosleđuje niz ranika, a ona ispisuje sve podatke za svakom od radnika sa maksimalnom platom.

1. Kreirati kalsu ***Autobus*** koja ima podatke o *registarskom broju* autobusa i o *broju sedišta* u autobusu.  
   Napraviti konstruktor kome se prosleđujeu odgovarajuće vrednosti.  
   Napraviti odgovarajuće getere i setere.

Napraviti metodu koja ispisuje podatke o autobusu.

Napraviti niz autobusa.

Napraviti funkciju ***ukupnoSedista*** kojoj se prosleđuje niz autobusa a koja određuje i vraća koliko ukupno sedišta sadrže svi autobusi koji se nalaze u nizu autobusa.

Napraviti funkciju ***maksSedista*** koja ispisuje podatke o autobusu koji ima najveći broj sedišta.

Napraviti funkciju ***ljudi*** kojoj se prosleđuje broj ljudi i niz autobusa a funkcija vraća true ukoliko je moguće toliko ljudi smestiti u autobuse, u suprotnom vraća false.

1. Kreirati klasu **FMRadio** koja sadrđi atribute o *stanici, frekvenciji* i *jačini tona*.

Klasa treba da sedrži odgovarajući konstruktor, getere i setere koji zadovoljavaju uslov da se frekvencija nalazi u opsegu od 87.5MHz do 108MHz. Jačina zvuka treba da se nalazi u opsegu od 0 do 20.

Napraviti metodu ***promeniZvuk*** kojoj se prosleđuje simbol plus ili minus i koja po pozivu jačinu zvuka menja isključivo za 1 stupanj. Ukoliko je prosleđen simbol + povećati jačinu zvuka za 1, u suprotnom smanjiti jačinu zvuka za 1. Voditi računa o tome da jačina zvuka ne sme biti manja od 0 i veća od 20 (Ova metoda ništa ne vraća, samo promeni vrednost zvuka).

Napraviti metodu ***frekvencijaIskljuci*** koja vrednost funkcije postavlja na najmanju moguću frekvenciju (na 87.5MHz).  
(I ova metoda takođe ništa ne vraća, samo postavi vrednost frekvencije)

Napisati funkciju ***srednjaFrekvencija*** kojoj se prosleđuje niz radio stanica, a ona vraća koja je prosečna frekvencija prosleđenih radio stanica.

Napisati funkciju ***najbliziMaks*** kojoj se prosleđuje niz radio stanica, a funkcija vraća onu radio stanicu čija je frekvencija najbliža maksimalnoj frekvenciji.

Testirati rad napravljenih metoda.

1. Kreirati klasu **Krug** koja ima atribut za poluprečnik kruga.  
   Napraviti odgovarajući konstruktor, getere i setere.  
   Napraviti metodu ***obimKruga*** koja izračunava i vraća obim kruga.  
   Napraviti metodu ***povrsinaKruga*** koja izračunava i vraća površinu kruga.

Napraviti funkciju ***upisaniKrug*** kojoj se prosleđuju dva objekta tipa krug, a ona vraća *true* ukoliko je u prvi prosleđeni krug moguće upisati drugi prosleđeni krug. U suprotnom funkcija vraća false.  
\* Napisati funkciju ***rastuci*** koja vraća *true* ukoliko se svaki krug može upisati u svog sledbenika. Ukoliko ma jedan krug ne može da se upiše u sledeći krug, vratiti *false*.

1. Napraviti klasu ***Cipela*** koja sadrži atribute: naziv, broj cipele i niz prepravki koje treba obaviti na cipeli.  
   Napraviti odgovarajući konstruktor, getere i setere.  
   Napraviti metodu ***zalepi*** koja po pozivu na ekranu ispisuje *“Zalepljena cipela”*.

Napraviti metodu ***prosiri*** koja po pozivu na ekranu ispisuje *“Prosirena cipela”*.

Napraviti metodu ***usij*** koja po pozivu na ekranu ispisuje *“Usivena cipela”*.

Napraviti funkciju ***radi*** kojoj se prosledjuje jedna cipela (jedna instanca klase Cipela) i koja proverava da li je neka u nizu prepravki sadrži string zalepi, proširi ili ušij. Svaki put kada naiđe na prepravku (element u nizu prepravki) zalepi poziva se metoda *zalepi*, kada naiđe na prepravku prosiri poziva se metoda prosiri, kada naiđe na prepravku usij poziva se metoda usij, u suprotno se na ekranu ispisuje “Nepoznata prepravka”.

1. (Malo teži od ostalih zadataka)  
   Kreirati klasu **Student** koja ima atribute: *jmbg, ime, prezime, nizOcena i presecna ocena*.  
   Prilikom kreiranja konstruktora, prosledjuje se jmbg, ime, prezime i nizOcena, a automatski se izračuna prosečna ocena ovog studenta.   
   Takođe, voditi računa o tome da svaka od unetih ocena može biti isključivo ceo broj i to 6, 7, 8, 9 ili 10.  
   JMBG studenta mora biti string koji sadrži 13 karaktera.  
   Izračunata prosečna ocena se dodeli atributu *prosecna ocena*.

Napraviti i odgovarajuće getere i setere.

Napraviti metodu ***ispis*** koja ispisuje sve podatke o jednom studentu.

Napraviti niz studenata i koristiti ga u predstojećim funkcijama.  
Napraviti funkciju ***imePrezime*** kojoj se prosleđuje ime i prezime, a funkcija vraća broj studenata sa tim imenom i prezimenom.

Napraviti funkciju ***srednjaOcena*** koja ispisuje koja je preosečna ocena svih ocena studenata.

Napraviti funkciju ***osrednji*** koja ispisuje studenta koji ima prosek najbliži srednjoj oceni svih studenata.

Napraviti ocenu ***najmanjaOcenaNajboljeg*** koja određuje najboljeg studenta i na ekranu ispisuje njegovu najslabiju ocenu.  
Napisati funkciju ***najmanjaOcena*** koja vraća koja je najmanja ocena, uzimajući u obzir ocene svih ocena koje su studenti dobijali.

Napisati funkciju ***najcescaOcena*** koja vraća ocenu koju su studenti najčešće dobijali.

Napisati funiciju ***rodjeniGodine*** kojoj se prosleđuje godina, a funkcija na ekranu ispisuje sve studente koji su rodjeni te godine. Godinu rođenja preuzeti (odrediti) iz JMBG-a studenta.

1. Kreirati klasu Plivac koja sadrži ime, godinu rodjenja i najbolji rezultat ove godine.  
   Napraviti odovarajuće konstruktore, getere i setere.  
   Napraviti metodu ***ispisi*** koja ispisuje sve podatke o plivaču.  
   Napraviti funkciju

Napraviti funkciju ***topTen*** koja kao rezultat vraća niz od 10 najboljih rezultata ove godine.

Napraviti funkciju ***norma*** kojoj se prosleđuje norma i niz plivača, a funkcija vraća niz onih plivača koji ispunjavaju ovu normu.

Napisati funkciju ***normaNajmladji*** koja treba da na ekranu ispiše sve podatke o najmlađem igraču koji zadovoljava normu koja je zadata prethodnom funkcijom.

1. Kreirati kalsu **Automobil** koji ima podatke o *maksimalnoj brzini, minimalnoj brzini i trenutnoj brzini*.   
   Pri stvaranju novog automobila (u konstruktoru) zadajete minimalnu vrednost brzine, maksimalnu vrednost brzine i trenutnu vrednost brzine.

Napraviti odgovarajuće setere i getere.

Napraviti metodu  ***ubrzaj*** kojoj se prosledjuje vrednost za koliko treba uvećati trenutnu brzinu. Ukoliko nije moguće izvršiti toliko ubrzanje, trenutnu brzinu postaviti na maksimalnu vrednost brzine, u suprotnom uvećati za traženu vrednost.

Napraviti metodu ***uspori*** kojoj se prosledjuje vrednost za koliko treba smanjiti trenutnu brzinu. Ukoliko nije moguće izvršiti toliko usporenje, trenutnu brzinu postaviti na minimalnu vrednost brzine, u suprotnom umanjiti brzinu za traženu vrednost.