

4 Apstraktne klase

Napomena za sve zadatke:

- Možete dodavati atribut koji će vam omogućiti da efikasnije uradite program, ali ne smeju da budu javni
- Možete dodati sve potrebne javne metode koje će obezbediti da program radi (uključujući konstruktore i destruktore);
- Ne smete unositi podatke u telu konstruktora (scanf, cin, ...)
- Izbegnite na svaki način dupliranje koda
- Unos test podataka mora da bude iz fajla, a ne sa standardnog ulaza, osim ako u zadatku nije bas naglašeno. Štampanje na standardni izlaz je u redu, osim ako su zadatku eksplicitno ne traži drugačije
- Ukoliko se traži testiranje ponašanja nizova, nizovi treba da imaju makar po 5 elemenata
- Ukoliko se traži kreiranje nekoliko objekata, to znaci najmanje 4
- Objekte koje kreirate u glavnom programu, kreirati u dinamičkoj zoni memorije, osim ako nije suprotno naglašeno

Zadatak 0:

Na programskom jeziku C++ kreirati klasu **Skladiste** koja od privatnih atributa sadrži maksimalni kapacitet skladišta u kg (double), broj elemenata niza i niz pointera na objekte klase **Element**. Klasa sadrži i sledeće javne funkcije:

- konstruktor koji postavlja maksimalni kapacitet, broj elemenata niza i zauzima potreban prostor u memoriji,
- destruktore,
- funkciju za dodavanje jednog elementa u niz ako time ukupna količina materijala ne prevazilazi maksimalnu vrednost,
- funkciju za određivanje ukupne količine materijala u skladištu,
- funkciju koja prikazuje listu šifara svih materijala koji su uskladišteni (svaku šifru štampati po jednom),
- prijateljsku operatorsku funkciju << za prikaz liste elemenata u skladištu uređenu u rastući niz po šifri materijala.

Kreirati klasu **Element** koja od zaštićenih atributa sadrži šifru (int), šifru materijala (int) i količinu (float), a od javnih funkcija:

- funkciju koja vraća šifru materijala,
- funkciju koja vraća količinu materijala,
- čistu virtuelnu funkciju za prikaz podataka o elementu,
- operatorsku funkciju za upoređivanje dva elementa (po šifri).

Kreirati klase **Paket** i **Silos** koje su javno izvedene iz klase **Element**. Obe klase treba da imaju privatni atribut lot (niz karaktera). Sem toga klasa **Paket** treba da ima i polje tip (char*), a klasa **Silos** treba da ima maksimalnu kolicinu materijala. Obe klase treba da imaju još i:

- konstruktor koji postavlja sve attribute,
- operator >> za unos podataka o paleti.

Obezbediti da tip paketa mora da počne slovom, a da silos ne može da prihvati veću količinu od definisane maksimalne količine.

U funkciji main napraviti skladište, učitati iz unapred pripremljenog fajla po 1009 paketa i silosa i dodati ih skladištu, a zatim odrediti ukupnu količinu materijala u skladištu. U tekstualni fajl upisati listu šifara svih materijala koji se javljaju u skladištu, a na standardni izlaz i prikazati jedinstvenu listu svih paketa i silosa leksikografski uređenih po lotu.

Zadatak 1:

Na programskom jeziku C++ kreirati klasu **Put** koja od privatnih atributa sadrži niz pointera tipa **Element**. Klasa sadrži i sledeće javne funkcije:

- konstruktor koji postavlja veličinu niza i zauzima potreban prostor u memoriji,
- destruktor,
- funkciju za postavljanje objekta tipa **Element** na određeno mesto u nizu,
- funkciju koja prikazuje broj elemenata po svakom od tipova koji postoje u nizu,
- prijateljsku operatorsku funkciju << za prikaz sadržaja niza.

Kreirati apstraktnu klasu **Element** koja od zaštićenih atributa sadrži tip (int) i od javnih sledeće funkcije:

- funkciju koja vraća tip,
- čistu virtuelnu funkciju koja vraća vrednost specifične vrednosti za element (za elemente zemlja nadmorsku visinu, za elemente voda dubinu),
- čistu virtualnu funkciju za štampanje podataka o elementu.

Kreirati klasu **Zemlja** koja je javno izvedena iz klase **Element**, koja od privatnih atributa sadrži nadmorsku visinu (int) i od javnih metoda sadrži:

- konstruktor koji tip postavlja na 1,
- operator >> za unos podataka o elementu.

Kreirati klasu **Voda** koja je javno izvedena iz klase **Element**, koja od privatnih atributa sadrži dubinu (float) i od javnih metoda sadrži:

- konstruktor koji postavlja tip na 2,
- operator >> za unos podataka o elementu.

U funkciji main napraviti **Put** od 2018 elementa, na osnovu podataka iz unapred pripremljenog fajla. Po 1009 elemenata kreirati kao objekte **Zemlja** i **Voda**. Prikazati sadržaj puta i prikazati koliko je prosečna nadmorska visina na putu (za polja tipa **Voda** uzeti da je nadmorska visina 0).

Zadatak 2:

Na programskom jeziku C++ kreirati klasu **Kalkulator** koja od privatnih atributa sadrži akumulator (int), naziv (char*) i niz pointera tipa **Operacija** i koji treba da funkcioniše kao stek. Klasa sadrži i sledeće javne funkcije:

- konstruktor koji postavlja naziv, broj elemenata niza i zauzima potreban prostor u memoriji,
- destruktor,
- funkciju koja postavlja akumulator na 0 i briše stek operacija,
- funkciju koja prikazuje stanje akumulatora,
- funkciju **Do(Operacija& novaOperacija)** koja dodaje *novaOperacija* u stek i izvršava je tako što se kao op1 prosleđuje vrednost akumulatora a kao op2 vrednost drugog operanda iz *novaOperacija*. Rezultat se smešta u akumulator.

- funkciju Undo koja uzima zadnju operaciju sa steka i izvršava njenu suprotnu operaciju gde je op1 vrednost akumulatora a op2 vrednost drugog operanda iz zadnje operacije.

Kreirati apstraktnu klasu **Operacija** koja od zaštićenih atributa sadrži naziv (char*), drugiOperand (int) i suprotnaOperacija (Operacija*):

- funkciju koja vraća suprotnu operaciju,
- funkciju koja vraća drugi operand,
- funkciju za prikaz podataka o operaciji (naziv, drugi operand i naziv suprotne operacije),
- čistu virtuelnu funkciju DoOperation(int op1, int op2) čiji je povratni tip int.

Kreirati klasu **Sabiranje** koja je javno izvedena iz klase **Operacija** i d javnih metoda sadrži:

- konstruktor koji postavlja sve attribute.

Kreirati klasu **Oduzimanje** koja je javno izvedena iz klase **Operacija** i od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute.

Kreirati klasu **Množenje** koja je javno izvedena iz klase **Operacija**. Od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute.

Kreirati klasu **Deljenje** koja je javno izvedena iz klase **Operacija**. Od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute.

U funkciji main napraviti Kalkulator sa baferom za 2020 operacija. Učitati po 505 sabiranja, oduzimanja, množenja i deljenja iz unapred pripremljenog fajla. Izvršiti sve računске, a onda primeniti par undo operacija. Prikazati sadržaj akumulatora nakon svake izvršene operacije.

Zadatak 3:

Kreirati klasu **Process** koja od zaštićenih atributa sadrži šifru (int), stanje (int, može da uzme vrednosti 0 – kreiran, 1 – startovan i 2 - zaustavljen) i tip (int) a od javnih funkcija sadrži:

- funkciju koja vraća šifru,
- funkciju koja vraća stanje,
- funkciju za prikaz šifre, stanja i tipa procesa,
- čistu virtuelnu funkciju za startovanje procesa koja kao parametar ima jedan niz karaktera - void Start(char* parameter),
- čistu virtuelnu funkciju za zaustavljanje procesa – void Stop().

Nakon kreiranja objekta tipa Process, objekat je u stanju 0. U stanje 1 prelazi nakon što se startuje pozivom odgovarajuće funkcije, a u stanje 2 nakon što se pozove funkcija za njegovo zaustavljanje.

Kreirati klasu **PublicProcess** koja je javno izvedena iz klase **Process**, koja od privatnih atributa sadrži ime korisnika koji ih je kreirao (char* creatorName). Od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute.

Kod objekata ove klase parametar tip uvek ima vrednost 0. Prilikom startovanja procesa prosleđuje se kao parametar funkcije Start ime korisnika koji startuje proces. Ako je ime korisnika koji inicira startovanje PublicProcess-a (userName) različito od vrednosti creatorName, creatorName treba da uzme vrednost koju ima userName, dok proces, u svakom slučaju, treba da pređe u stanje 1.

Kreirati klasu **PrivateProcess** koja je javno izvedena iz klase **Process**, koja od privatnih atributa ima kod (char* code) i od javnih metoda sadrži konstruktor koji postavlja sve attribute.

Kod objekata ove klase parametar tip uvek ima vrednost 1. Ove procese može da startuje samo korisnik koji zna njihov kod i koji prosleđuje kod kao parametar funkcije za startovanje, a mogu da pređu u stanje 2 samo ako se trenutno nalaze u stanju 1.

Na programskom jeziku C++ kreirati klasu **ProcessQueue** koja od privatnih atributa sadrži maksimalni broj elemenata po nizu i dva niza pointera na objekte klase **Process**. Jedan niz predstavlja aktivne procese a drugi procese na čekanju. Klasa sadrži i sledeće javne funkcije:

- konstruktor koji postavlja broj elemenata nizova i zauzima potreban prostor u memoriji,
- destruktor,
- funkciju koja za dodavanje pointera na tip **Process** u specificirani niz ako u njemu postoji slobodno mesto,
- funkciju koja iz nizova izbacuje sve procese u stanju 2,
- funkciju koja startuje sve procese iz oba niza koji su u stanju 0,
- funkciju koja zaustavlja sve procese iz oba niza koji su u stanju 1,
- operatorsku funkciju + koja spaja dva objekta tipa **ProcessQueue** u jedan,
- funkciju koja prikazuje sadržaj oba niza.

U funkciji **main** testirati sve funkcije svih klasa.

Zadatak 4:

Aždaja treba da pređe put koji se proteže kroz izvestan broj gora, mora i močvara. Za svaki od segmenata puta se može odrediti koeficijent težine i koeficijent lepote. Takođe, svaki od segmenata puta ima i naziv.

More se dodatno opisuje površinom (P), stepenom plavetnila (S) i maksimalnom snagom vetra (MSV , ceo broj u opsegu 0 do 12). Za more, koeficijent lepote se računa kao $L = MSV * S / P$, a koeficijent težine kao $MSV * P / (L - 1)$.

Gora se dodatno opisuje površinom (P), dužinom puta kroz nju (D), relativnom razlikom nadmorske visine između najniže i najviše tačke (V) i stepenom inverzne pošumljenosti (označava se sa B , i on je razlomljeni broj u opsegu 0.01 do 1). Za gore, koeficijent težine se računa kao $T = B * D / (D + P + V + D/P)$. Koeficijent lepote je $(1 - B) * V$.

Močvara se opisuje površinom (označava se sa P , i meri u km^2), stepenom žitkosti mulja (označava se sa Z , i predstavlja razlomljeni broj u opsegu 0 do 0.99). Za močvaru se koeficijent težine računa kao $T = (1 - Z) * (1 - 1 / (S * P))$. Koeficijent lepote je $1 - T$.

Put kojim aždaja treba da prođe se sastoji od 777 gora, 777 mora i 777 močvara.

Omogućite aždaji da može da učitava sve segmente puta iz prethodno pripremljenog fajla, zatim da ih sortira po koeficijentu težine, pa koeficijentu lepote. Omogućite joj i da snimi u fajl niz nakon oba sortiranja kako bi mogla da se odluči kako da isplanira svoj put. Kada štampate segmente puta, vodite računa da budu odštampani svi atributi svakog elementa, uključujući i koeficijente težine i lepote.

Zadatak 5:

Na programskom jeziku C++ kreirati klasu **Workflow** koja od privatnih atributa sadrži dužinu niza i niz pointera tipa **Step**. Klasa **Workflow** sadrži i sledeće javne funkcije:

- konstruktor koji postavlja dužinu niza i zauzima potreban prostor u memoriji,
- destruktor,
- konstruktor za kopiranje,
- funkciju *InsertAt* za dodavanje jednog elementa u niz na mesto sa prosleđenim indeksom,
- funkciju *RemoveAt* koja izbacuje iz niza element na određenom mestu,

- funkciju *GetSpecificSteps* (vraća niz pointera tipa *Step*) koja vraća niz pointera na sve korake određenog tipa (tip se prosleđuje kao parametar funkcije, i može da bude „Activity“ ili „State“),
- prijateljsku operatorsku funkciju << za prikaz svih elemenata u nizu.

Kreirati apstraktnu klasu **Step** koja od zaštićenih atributa sadrži naziv (*char**) i tip (*char**) i od javnih sledeće funkcije:

- funkciju koja vraća naziv objekta,
- čistu virtuelnu funkciju (*GetSpecAttribute*) koja treba da vraća podatak tipa *int*,
- virtuelnu funkciju za prikaz vrednosti atributa naziv, tip i vrednosti koju vrati funkcija *GetSpecAttribute*.

Kreirati klasu **Activity** koja je javno izvedena iz klase **Step** i koja od privatnih atributa ima indicaciju da li je obavezna ili nije (ceo broj, može da uzme vrednosti 0 i 1). Od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute (vrednost atributa tip postavlja na „Activity“),
- operator >> za unos podataka o objektom tipa *Activity*.

Kreirati klasu **State** koja je javno izvedena iz klase **Step** i koja od privatnih atributa sadrži prioritet (ceo broj). Od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute (vrednost atributa tip postavlja na „State“),
- operator >> za unos podataka o objektu tipa *State*.

U funkciji *main* napraviti Workflow W1 od 2018 koraka. Definirati polovinu koraka kao aktivnosti (tipa *Activity*) a polovinu kao stanja (tipa *State*) zatim pozvati sve funkcije iz klase *Workflow*.

Zatim kreirati novi Workflow W3 u koji treba prebaciti sve korake iz W1 koji su tipa „Activity“.

Na kraju kreirati još dva Workflow objekta i testirati operator dodele i konstruktor za kopiranje.

Zadatak 6:

Definisati klasu **FileSystemElement** koja od zaštićenih atributa ima ime (*char**), ekstenziju (*char**) i pokazivač na roditeljski element (*FileSystemElement**), a od javnih funkcija ima:

- konstruktor koji postavlja ime i ekstenziju,
- destruktor,
- funkciju koja postavlja roditeljski element,
- čistu virtuelnu funkciju tipa *char** koja vraća puno kvalifikovano ime,
- čistu virtuelnu funkciju *PrintList* tipa *void* i bez parametara.

Kreirati klasu **Folder** koja javno nasledjuje **FileSystemElement**. Ekstenzija kod foldera je uvek prazan string i ne može se predefinisati, a puno kvalifikovano ime čini konkatenacija imena roditeljskog (i rekurzivno, svih njegovih roditelja) i samog foldera. Od dodatnih atributa folder ima niz pokazivača na objekte tipa **FileSystemElement** maksimalnog kapaciteta 2018 u kome čuva svoje podelemente.

Od dodatnih funkcija **Folder** ima:

- Metodu za dodavanje prosleđenog podelementa na prvo slobodno mesto u nizu. Kada se podelement a dodaje u folder b, folder b postaje direktni roditelj podelementu a.
- Metodu za uklanjanje podelementa sa mesta u nizu čiji je indeks prosleđen. Kada se podelement uklanja iz niza pointer na njegov roditeljski element više ne ukazuje ni na šta.

Kreirati klasu **File** koja javno nasleđuje **FileSystemElement**. Puno kvalifikovano ime kod fajlova čini puno kvalifikovano ime roditeljskog foldera konkaterinano sa imenom fajla i njegovom ekstenzijom.

U slučaju kada se **PrintList** zove nad nekim od foldera, štampati sve podelemente foldera zajedno sa svim njihovim podelementima (rekurzivno).

U slučaju kada se **PrintList** zove nad nekim od fajlova u listu treba da se nađu svi fajlovi koji imaju isti roditeljski element.

U glavnom programu napisati primer korišćenja implementiranih klasa i metoda. Testirati sve metode.

Zadatak 7:

Na programskom jeziku C++ kreirati klasu **Preduzece** koja od privatnih atributa sadrži naziv (char*), budžet (int), niz pointera na objekte klase **Radnik**, maksimalni broj elemenata niza i broj elemenata niza. Klasa sadrži i sledeće javne funkcije:

- konstruktor koji postavlja naziv i maksimalni broj elemenata niza, broj elemenata niza postavlja na 0 i zauzima potreban prostor u memoriji,
- destruktor koji oslobađa zauzeti prostor u memoriji,
- funkciju za dodavanje jednog radnika u niz ako time ukupni broj elemenata ne prevazilazi maksimalnu vrednost koja ima kao parametar i godinu zapošljenja radnika,
- funkciju za brisanje radnika na osnovu JMBG koja uklanja zadatog radnika iz niza i postavlja njegovu godinu zapošljenja na 0,
- funkciju za određivanje ukupne svote novca potrebne za isplatu zarade radnicima,
- funkciju koja određuje da li je firma rentabilna (budžet preduzeća veći od svote novca potrebne za isplatu zarade radnicima),
- operatorsku funkciju ++ koja povećava osnovicu plate svih radnika za 10%,
- operatorsku funkciju -- koja smanjuje osnovicu plate svih radnika za 10%,
- prijateljsku operatorsku funkciju << za prikaz liste radnika u preduzeću uređenu u rastući redosled po godinama starosti.

Kreirati klasu **Radnik** koja od zaštićenih atributa sadrži JMBG (int), ime i prezime (char*) i godinu zapošljenja (int) i godine starosti (int) i osnovicu plate, a od javnih funkcija:

- funkciju koja vraća JMBG radnika,
- funkciju koja vraća ime i prezime radnika,
- funkciju koja vraća godinu zapošljenja radnika,
- čistu virtuelnu funkciju za određivanje plate radnika,
- operatorsku funkciju ++ koja povećava osnovicu plate radnika za 10%,
- operatorsku funkciju -- koja smanjuje osnovicu plate radnika za 10%,
- operatorsku funkciju za upoređivanje dva elementa (po godini starosti),
- čistu virtuelnu funkciju za prikaz podataka o radniku.

Kreirati klasu **Developer** koja je javno izvedena iz klase **Radnik**, koja od privatnih atributa sadrži tip („programer“, „projektant“, „tester“) i prekovremeni sati (int). Od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute klase,
- funkciju za određivanje plate radnika po formuli: $25 * osnovica + 1500 * godina zapošljenja + 1000 * prekovremeni sati$,

- operatorsku funkciju >> za unos podataka o radniku.

Kreirati klasu **Menager** koja je javno izvedena iz klase **Radnik**, koja od privatnih atributa sadrži tip („finansijski“, „tehnički“ i „marketinški“). Od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute klase,
- funkciju za određivanje plate radnika po formuli: $30 * osnovica + 1000 * godina zapošljenja$,
- operator >> za unos podataka o radniku.

U funkciji main napraviti preduzeće, napraviti po 1009 developera i menadžera i dodati ih preduzeću. Zatim proveriti da li je firma rentabilna i ukoliko jeste povećati osnovice svim radnicima za 10%, a u suprotnom ih smanjiti za 10%. Odštampati podatke o preduzeću i njegovim radnicima.

Zadatak 8:

Kreirati klasu **Zadatak** koja ima privatni atribut kod (int) i stepen težine (int).

Od javnih funkcija ima:

- konstruktor bez parametara,
- destruktor,
- funkciju koja postavlja kod i stepen težine,
- funkciju koja vraća kod,
- funkciju koja vraća stepen težine.

Kreirati klasu **Heroj** koja od zaštićenih atributa sadrži ime (char*), vek u kome je rođen (int) i niz od 17 pointera na objekte tipa **Zadatak**. Od javnih funkcija klasa sadrži:

- konstruktor koji postavlja sve parametre,
- destruktor,
- operator dodele,
- operator [] koji vraća onaj zadatak čiji je indeks prosleđen,
- funkciju koja vraća ime,
- funkciju koja vraća broj dodeljenih zadataka,
- čistu virtuelnu funkciju za dodavanje novog zadatka u red,
- čistu virtuelnu funkciju za prikaz vrednosti svih privatnih i zaštićenih članova klase.

Kreirati klasu **MitskiHeroj** koja je javno izvedena iz klase **Heroj**, i koja od privatnih atributa ima broj mitova u kojima se heroj pominje i minimalnu težinu zadatka koju prihvata i od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute.

Mitski heroj može da prihvati samo one zadatke čija je težina veća ili jednaka definisanoj minimalnoj težini.

Kreirati klasu **BitskiHeroj** koja je javno izvedena iz klase **Heroj**, koja nema dodatne parametre, a od javnih metoda sadrži:

- konstruktor koji postavlja sve attribute.

Bitski heroj može da prihvati samo one zadatke koji su težine 0 ili 1.

U glavnom programu kreirati niz od 20 pointera na objekte klase Heroj. Za svaki pointer sa parnim indeksom kreirati objekat tipa MitskiHeroj, a za svaki pointer sa neparnom vrednošću indeksa objekat tipa BitskiHeroj. Kreirati zatim niz od 30 zadataka i dodeliti ih herojima. Na kraju prikazati imena svih heroja koji su završili po više od 3 zadatka.

Zadatak 9:

Vitez treba da vežba za turnir i u okviru tog procesa treba da se bori sa izvesnim brojem aždaja, mantikora i belih zečeva. Svi protivnici se opisuju imenom, koeficijentom borbene sposobnosti (B) i koeficijentom magične privlačnosti (M), i posebnim dodatnim atributima koji su svi razlomljeni brojevi u opsegu 0 do 1.

Aždaje se dodatno opisuju sjajnošću krljušti (S), simetrijom krila (K), temperaturom vatre (T) i snagom udarca (U). Za njih se koeficijenti računaju kao $B = T * (1 + K) * U$, i $M = S * K$.

Mantikori se dodatno opisuju intenzitetom crvene boje (I), simetrijom rogova (K), stepenom efikasnosti otrova (O) i efektivnom snagom ugriza (U). Za njih se koeficijenti računaju kao: $B = 0.5 * (U + O) * K$, a $M = I * (K + O + U) / 3$.

Beli zečevi se dodatno opisuju intenzitetom bele boje (C) i nivoom znanja korišćenja samurajskog kratkog mača (wakizashi skill level, W)). Za njih je $B = W$ i $M = C$.

Skup neprijatelja protiv kojih vitez treba da vežba sastoji se od 777 aždaja, 777 mantikora i 777 belih zečeva.

Omogućite vitezima da može da sortira sve segmente puta po koeficijentu B ili po koeficijentu M, odštamajte mu oba predloga da bi mogao da se odluči kako da isplanira svoj trening. Kada štampate, vodite računa da budu odštampani svi atributi svakog protivnika, uključujući i koeficijente B i M.