

5 Templejtske klase

Napomena za sve zadatke:

- Možete dodavati atribute 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 templejtsku klasu Set koja modelira skup sa elementima proizvoljnog tipa. Privatni atributi klase su broj elemenata skupa i dinamički niz elemenata, a kao javne (public) sledeće funkcije članice:

- default konstruktor koji postavlja broj elemenata skupa na 0,
- konstruktor za postavljanje broja elemenata skupa i zauzimanje neophodnog prostora u memoriji,
- destruktore,
- inline funkciju koja vraća broj elemenata skupa,
- funkciju koja ispituje da li zadati element pripada skupu,
- operatorski funkciju * koja odredjuje presek 2 skupa.

Implementirati i prijateljske operatorske funkcije:

- >> - za učitavanje elemenata skupa,
- << - za upis elemenata skupa.

Takođe, kreirati klasu Movie koja od privatnih podataka ima:

- naziv filma
- puno ime reditelja
- trajanje filma

a od javnih sledeće metode:

- Default konstruktor koji postavlja sve podatke na nulte vrednosti
- Destruktor
- Operator = kojidodeljuje jedan objekat klase Movie drugom

- Operator == koji poredi dva filma
Definisati i prijateljske funkcije:
- Operator << koji štampa naziv filma, puno ime reditelja i trajanje filma.
- Operator >> koji učitava sve attribute filma.

U glavnom programu specijalizovati templejt i testirati sve njegove metode.

- Za podatke tipa double
- Za klasu Movie

Zadatak 1. Napisati templejtsku klasu koja modelira niz sačinjen od proizvoljnog broja elemenata proizvoljnog tipa. Kao privatne atribut ova klasa treba da ima niz elemenata zapamćen u statičkoj zoni memorije. Kao javne, klasa treba da ima sledeće funkcije:

- Operator[] za pristup elementima niza
- Metodu koja vraća broj elemenata niza
- Funkciju za učitavanje elemenata niza sa standardnog ulaza
- Funkciju za štampanje sadržaja niza standardni izlaz.
- Funkciju za uredjivanje elemenata niza u neopadajući redosled.

Implementirati i klasu Student koja će kao privatne članove imati ime i prezime studenta, broj indeksa i prosečnu ocenu sa do sada položenih ispita, a kao javne (public) sledeće funkcije članice:

- Operatorski funkciju > koja poredi dva studenta prema prosečnoj oceni
- Operatorsku funkciju = koja dodeljuje jedan objekat klase Student drugom.

Takođe implementirati i prijateljske operatorske funkcije:

- >> za učitavanje podataka o studentu,
- << za štampanje podataka o studentu.

U funkciji main specijalizovati templejt i testirati sve njegove metode.

- Za podatke tipa double
- Za klasu Student

Zadatak 2. Kreirati templejtsku klasu Poligon koja će kao privatni atribut imati statički vektor tacaka (pri cemu su tip tacke i broj tacaka argumenti templejta), a kao javne (public) sledeće funkcije članice:

- funkciju koja izračunava obim poligona,
- prefiksnu operatorski funkciju -- koja preslikava poligon simetrično u odnosu na x osu (tako što istu operaciju izvrši nad svim tačkama),

- postfiksnu operatorsku funkciju – koja preslikava poligon simetrično u odnosu na y osu (tako što istu operaciju izvrši nad svim tačkama).

Takođe implementirati i prijateljske operatorske funkcije:

- >> za učitavanje koordinata temena,
- << za upis koordinata temena.

Kreirati i klasu DekartovaTačka koja od privatnih članova ima x i y koordinate tačke, a od javnih sledeće metode:

- Default konstruktor koji postavlja sve podatke na nulte vrednosti
- Funkciju za izračunavanje rastojanja između dve tačke.
- Prefiksni operator – koja preslikava tačku simetrično u odnosu na x-osu.
- Postfiksni operator – koja preslikava tačku simetrično u odnosu na y-osu.

Definisati i prijateljske funkcije:

- Operator << koji štampa koordinate tačke.
- Operator >> koji učitava atribut tačke

Kreirati i klasu PolarnaTačka koja modelira tačku u polarnom koordinatnom sistemu (u kojem je tačka predstavljena svojim rastojanjem od koordinatnog početka i uglom koji zaklapa sa polarnom osom koju treba tretirati kao x osu). U klasi implementirati sve funkcije koje postoje i u klasi DekartovaTačka.

U glavnom programu specijalizovati template za jedan i drugi tip tačke i testirati sve njegove metode.

Zadatak 3. Kreirati template klasu Image koja će kao privatni član imati kvadratnu matricu piksela proizvoljne dimenzije i proizvoljnog tipa, a kao javne (public) sledeće funkcije članice:

- funkciju koja invertuje sliku (pri čemu ona svaki element invertuje primenom operatora !),
- funkciju za učitavanje slike sa standardnog ulaza,
- funkciju za štampanje slike na standardni izlaz.

Kreirati i klasu Color čiji su privatni atributi R, G i B komponenta boje, a javne funkcije:

- operatorska funkcija ! koja komplementira sve komponente boje (u intervalu 0-255),
- operatorsku funkciju =.

Definisati i prijateljske funkcije:

- Operator << koji štampa boju,
- Operator >> koji učitava sve komponente boje.

U funkciji main krierati objekte:

- crnoBelaSlika (u kojoj su pikseli predstavljeni tipom int i uzimaju vrednosi 0 i 1),
- slikaUBoji (u kojoj su pikseli predstavljeni klasom Color)

i nad oba objekta testirati sve funkcije klase Image.

Zadatak 4. Napisati templejtsku klasu koja modelira niz sačinjen od elemenata proizvoljnog tipa. Kao privatne podatke ova klasa treba da ima niz elemenata proizvoljnog tipa i broj elemenata tog niza. Kao javne, klasa treba da ima sledeće funkcije:

- Default konstruktor koji zauzima prostor za 10 elemenata
- Konstruktor koji postavlja broj elemenata i zauzima prostor elemenete niza
- Destruktor
- Operator =
- Operator[] za pristup elementima niza
- Int GetSize() – metodu koja vraća broj elemenata niza
- Operatorsku funkciju + koja sabira sve elemente sa odgovarajućim indeksima. Dužina rezultatnog niza jednaka je dužini kraćeg od njih.

Takođe, kreirati klasu E3Point koja od privatnih podataka ima:

- X, Y i Z koordinate tačke
- naziv tačke

a od javnih sledeće metode:

- Default konstruktor koji postavlja sve podatke na nulte vrednosti
- Konstruktor sa tri argumentom koji vrednosti svih koordinata postavlja na prosleđene vrednost
- Destruktor
- Operator + koji sabira koordinate dve tačke i vrši konkatenciju njihovih naziva
- Operator = koji deodeljuje jedan objekat tipa E3Point drugom

Definisati i prijateljske funkcije:

- Operator << koji štampa koordinate i naziv tačke.
- Operator >> koji učitava koordinate i naziv tačke.

U glavnom programu specijalizovati templejt i testirati sve njegove metode.

- Za celobrojne podatke
- Za klasu E3Point

Zadatak 5. Napisati templejtsku klasu koja modelira matricu elemenata proizvoljnog tipa. Kao privatne podatke ova klasa treba da ima matricu elemenata proizvoljnog tipa i veličine obe dimenzije. Kao javne, klasa treba da ima sledeće funkcije:

- Default konstruktor koji zauzima prostor za 12×12 elemenata
- Konstruktor koji postavlja dimenzije matrice i zauzima prostor za njegove elemente
- Destruktor
- Operator =
- Funkciju `GetElement(int x, int y)` za čitanje vrednosti konkretnog elementa
- Funkciju `SetElement(int x, int y, T newValue)` za postavljanje nove vrednosti konkretnog elementa
- Funkcije `GetX` i `GetY` koje vraćaju veličine dimenzija
- Operatorsku funkciju `*` za množenje matrica

Takođe, kreirati klasu `Complex` koja od privatnih podataka ima:

- realni deo kompleksnog broja (`double`)
- imaginarni deo kompleksnog broja (`double`)

i sledeće javne metode:

- Default konstruktor koji postavlja sve podatke na nulte vrednosti
- Konstruktor koji vrednosti svih privatnih atributa postavlja na prosleđene vrednosti
- Destruktor
- Operatorsku funkciju `*` za množenje dva kompleksna broja
- Operatorsku funkciju `+` za abiranje dva kompleksna broja
- Operatorsku funkciju `=`

Definisati i prijateljske funkcije:

- Operator `<<` koji štampa kompleksni broj u formatu `Re + jIm.`
- Operator `>>` koji učitava attribute kompleksnog broja

U glavnom programu specijalizovati templejt i testirati sve njegove metode.

- Za celobrojne podatke
- Za klasu `Complex`

Zadatak 6. Napraviti šablonsku klasu ***Polinom*** u kojoj se koeficijenti pamte kao niz podataka proizvoljnog tipa. Polinom može biti proizvoljnog stepena, i taj podatak treba biti predstavljen kao netipski parametar templejta. Od javnih funkcija klasa sadrži:

- konstruktor
- funkciju koja sa standardnog ulaza učitava niz koeficijenata polinoma
- funkciju koja izračunava vrednost polinoma za dato `x` (`x` je istog tipa kao i koeficijenti)

Takođe, kreirati klasu `Complex` koja od privatnih podataka ima:

- realni deo kompleksnog broja (`double`)
- imaginarni deo kompleksnog broja (`double`)

a od javnih članova sledeće metode:

- Default konstruktor koji postavlja sve podatke na nulte vrednosti
- Konstruktor koji vrednosti svih privatnih atributa postavlja na prosleđene vrednosti
- Destruktor
- Operatorsku funkciju `+` za sabiranje dva kompleksna broja
- Operatorsku funkciju `*` za množenje dva kompleksna broja
- Operatorsku funkciju `=`

Definisati i prijateljske funkcije:

- Operator `<<` koji štampa kompleksni broj u formatu `Re + jIm`.
- Operator `>>` koji učitava attribute kompleksnog broja

U main funkciji kreirati po jedan realni i kompleksni polinom stepena 5, učitati njihove koeficijente sa standardnog ulaza i na standardni izlaz prikazati vrednosti polinoma za $x=2$.

Zadatak 7. Napraviti šablonsku klasu ***Buffer*** koja predstavlja niz elemenata proizvoljnog tipa, maksimalne dužine zadate kao netipski parametar šablona. Klasa ima sledeće funkcije:

- `getItem(int index)` vraća element na poziciji *index* u nizu
- `getFirst()` koja vraća prvi element liste
- `getLast()` koja vraća zadnji element liste
- `add` koja dodaje novi element u listi

Implementirati i klasu `Student` koja će kao privatne članove imati ime i prezime studenta, broj indeksa i prosečnu ocenu sa do sada položenih ispita, a kao javne (`public`) sledeće funkcije članice:

- Konstruktor bez argumenata,
- Konstruktor za kopiranje,
- Operatorsku funkciju `=` koja dodeljuje jedan objekat klase `Student` drugom.

Takođe implementirati i prijateljske operatorske funkcije:

- `>>` za učitavanje podataka o studentu,
- `<<` za štampanje podataka o studentu.

U main funkciji kreirati objekat klase `Buffer` koji radi sa karakterima i objekat klase `Buffer` koji radi sa studentima testirati sve metode klase `Buffer` za oba objekta.

Zadatak 8. Napraviti šablonsku klasu **Cursor** koja sadrži listu elemenata proizvoljnog tipa i proizvoljne maksimalne dužine (oba su parametri šablona). Klasa sadrži atribut *current* koji pamti poziciju kursora u nizu, a od javnih funkcija sadrži:

- `getCurrent()` vraća element na kome se trenutno nalazi kursor
- operator `--` koji pomera kursor za jednu poziciju ulevo u nizu, ukoliko nije početak niza
- operator `++` koji pomera kursor za jednu poziciju udesno u nizu, ukoliko nije kraj niza
- operator `-` koji postavlja kursor za n pozicija ulevo u nizu. Ukoliko je trenutna pozicija kursora manja od broja n , kursor će biti postavljen na početak niza
- operator `+` koji postavlja kursor za n pozicija udesno u nizu. Ukoliko od trenutne pozicije kursora do kraja niza ima manje od n pozicija, kursor će biti postavljen na kraj niza
- *add* koja dodaje novi element na kraj niza

Kreirati i klasu *Pravougaonik* koja ima privatne atribute : *duzinu* i *sirinu* i prijateljske operatorske funkcije za *ulaz* i *izlaz*.

U *main* funkciji napraviti po jedan objekat klase *cursor* dimenzije 20 koji radi sa karakterima i sa *pravougaonicima* i nad oba objekta testirati sve metode klase *Cursor*.

Zadatak 9. Napraviti šablonsku klasu **Bioskop** koja je realizovana kao matrica elemenata proizvoljnog tipa. Svaki element predstavlja jednog gledaoca u bioskopu. Dimenzije bioskopa su zadate kao netipski parametri šablona. Definirati sledeće metode:

- podrazumevani konstruktor koji postavlja elemente na 0
- metodu koja računa prosek godina gledalaca u bioskopu (koristeći metode elemenata)
- metodu koja računa prosečnu ocenu filma na osnovu mišljenja publike (koristeći metode elemenata)
- operator `()` koji vraća referencu na element na zadatoj lokaciji

Kreirati klasu **Gledalac** koja sadrži sledeće atribute:

- ime i prezime
- starost
- ocena filma
- prisutan

Ova klasa sadrži i sledeće metode:

- konstruktor

- destruktor
- metodu koja vraća ocenu filma
- metodu koja vraća starost
- metodu koja ispituje da li je gledalac prisutan

Definisati i operatore ulaza i izlaza kao prijateljske funkcije. U operatoru ulaza najpre učitati da li je gledalac prisutan, pa tek ako jeste učitati i ostale njegove atribute.

U main funkciji napraviti Bioskop sa elementima tipa Gledalac i proizvoljne veličine. Napraviti više objekata tipa Gledalac i rasporediti ih po bioskopu. Prikazati prosek godina posetilaca kao i prosečnu ocenu filma. Pri izračunavanjima treba uzeti u obzir samo prisutne gledaoce.