

II PARCIJALNI ISPIT IZ PREDMETA “TEHNIKE PROGRAMIRANJA” (GRUPA A)

VAŽNA NAPOMENA:

U svim klasama koje treba razviti, trivijalne metode koje se mogu implementirati u jednoj ili najviše dvije naredbe možete implementirati direktno unutar deklaracije klase, dok sve ostale metode trebate implementirati izvan deklaracije klase. Također, sve metode koje su po svojoj prirodi inspektori, obavezno deklarirajte kao takve.

Zadatak 1 (7 poena):

Razvijte apstraktnu baznu klasu koja predstavlja neki geometrijski lik. Jedini atributi (privatni) koje ova klasa treba sadržavati su naziv lika (tipa “string”) i interni cjelobrojni brojač, čija će uloga biti objašnjena kasnije. Interfejs klase treba da sadrži sljedeće elemente:

- Konstruktor sa jednim parametrom, koji postavlja naziv lika na vrijednost zadanu parametrom i inicijalizira pomenuti brojač na jedinicu.
- Trivijalne metode bez parametra, koje vraćaju naziv lika odnosno vrijednost brojača pohranjenog u klasi.
- Čisto virtualnu metodu bez parametara koja je namijenjena za računanje obima lika (ova metoda će biti neimplementirana, dok će konkretne implementacije biti izvedene u klasama naslijedenim iz ove bazne klase).
- Preklopljeni operator “<<” za ispis podataka o liku na izlazni tok, koji ispisuje podatke o liku na sljedeći način (računanje obima izvodi se pozivom odgovarajuće virtuelne metode):

Lik: <naziv_lik> *Obim:* <iznos_obima>

- Preklopljene relacione operatore “<” i “>” koji vrše poredenje dva lika, pri čemu se smatra da je manji onaj lik koji ima manji obim.
- Preklopljeni unarni operator “++” koji povećava vrijednost internog brojača pohranjenog u liku za jedinicu. Potrebno je podržati samo prefiksnu verziju ovog operatora.

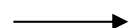
Iz ove bazne klase treba naslijediti tri nove klase koje predstavljaju krug, pravougaonik i trougao (trokut). Ove klase sadrže dodatne attribute koji specificiraju ove klase: poluprečnik za krug, dvije stranice za pravougaonik, te dužine tri stranice za trougao (trokut). Svi ovi atributi su realni brojevi. Interfejs ovih klasa sadrži sljedeće izmjene u odnosu na baznu klasu:

- Konstruktori ovih klasa imaju respektivno jedan, dva odnosno tri parametra, koji služe za inicijalizaciju odgovarajućih atributa. Pored toga, svi ovi konstruktori treba da inicijaliziraju i atribut koji čuva naziv lika (naslijeden iz bazne klase) na odgovarajući sadržaj.
- Metode za računanje obima u ovim klasama izvedene su u skladu sa tim kako se zaista računa obim za svaki od specifičnih likova.

Sve ostalo u ovim klasama nasljeđuje se iz odgovarajuće bazne klase.

Zadatak 2 (10 poena):

Razvijte kontejnersku klasu koja čuva kolekciju podataka o geometrijskim likovima. Pri tome, pretpostavite da se podaci o pojedinim likovima čuvaju u primjercima klase razvijenih u prethodnom zadatku (ovaj zadatak možete uraditi bez obzira da li ste uradili prethodni – dovoljno je samo da pretpostavite da klase sa navedenim svojstvima *postoje*). Kontejnerska klasa treba da sadrži pokazivač na prvi element dinamički alociranog niza pokazivača koji pokazuju na dinamički alocirane objekte koji sadrže stvarne podatke o pohranjenim likovima. Ova klasa još sadrži informaciju o broju pohranjenih likova, zatim informaciju o kapacitetu, odnosno maksimalnom broju likova koji se mogu pohraniti (ova informacija se čuva u konstantnom atributu), i konačno, pokazivač na brojač pristupa



klasi (on će se koristiti u konstruktoru kopije i destrukturu). Inače, svi atributi klase moraju biti u privatnoj sekciji klase. Interfejs klase treba da sadrži sljedeće elemente:

- a) Konstruktor sa jednim cjelobrojnim parametrom, koji predstavlja maksimalni broj likova koji se mogu pohraniti. Konstruktor treba da inicijalizira odgovarajuće attribute i da izvrši neophodnu alokaciju memorije. Pri tome je potrebno zabraniti da se ovaj konstruktor koristi za automatsku konverziju cjelobrojnog tipa u tip klase.
- b) Konstruktor kopije, koji prosto kopira sve attribute i povećava brojač pristupa klasi za jedinicu (tako da ćemo raditi sa plitkim kopijama). Međusobno dodjeljivanje primjeraka ove klase treba zabraniti.
- c) Destruktor, koji umanjuje brojač pristupa klasi za jedinicu, i ukoliko je on dostigao nulu, oslobađa svu memoriju koja je zauzeta tokom života primjeraka klase.
- d) Metode za dodavanje novog kruga, pravougaonika i trougla (trokuta) u kolekciju. Ove metode imaju onoliko parametara koliko je potrebno da se u potpunosti opiše razmatrani lik. Pored normalnih aktivnosti, sve ove tri metode treba da provjere da li među postojećim likovima već postoje likovi sa istim obimom kao lik koji se upravo dodaje, i ukoliko postoje, da poveća njihove interne brojače. Također, sve tri metode treba da bace izuzetak u slučaju da je kolekcija popunjena. Napomena: pošto u sve tri metode ima mnogo zajedničkih stvari, poželjno je definirati pomoćnu privatnu metodu koja radi sve zajedničke stvari, da izbjegnute dupliranje koda.
- e) Metodu koja vraća ukupan broj likova u kolekciji.
- f) Metodu koja vraća referencu na lik koji ima najveći obim (referenca se vraća sa ciljem da bi se mogao koristiti polimorfizam).
- g) Metodu koja vraća koliko ima krugova pohranjenih u kolekciji (preporučljivo je koristiti funkciju “count_if” iz biblioteke “algorithm”, uz definiranje odgovarajuće funkcije kriterija).
- h) Metodu koja sortira sve likove pohranjene u kolekciji u opadajući poredak po iznosu obima. Za tu svrhu, koristiti funkciju “sort” iz biblioteke “algorithm”, uz definiranje odgovarajuće funkcije kriterija.
- i) Preklopljeni unarni operator “!” , koji vraća logičku vrijednost “true” ako je kolekcija prazna, inače vraća “false”.
- j) Preklopljeni operator “<<” za ispis podataka o kolekciji na izlazni tok, koji ispisuje podatke o svim pohranjenim likovima (svaki lik u posebnom redu).
- j) Preklopljen operator indeksiranja “[]” koji omogućava da se pristupi i -tom pohranjenom liku, gdje je indeks i naveden kao parametar u uglastim zagradama. U slučaju da je indeks izvan opsega, treba baciti izuzetak. Ovaj operator treba da vrati konstantnu referencu na pohranjeni lik.

Zadatak 3 (3 poena):

Napišite glavni program koji će iz tekstualne datoteke “LIKOVI_A.TXT” pročitati podatke u likovima, upisati ih u kolekciju, sortirati u opadajući poredak po iznosu obima i ispisati podatke o svim likovima o kolekciji na ekran. Datoteka je organizirana tako da prvi red sadrži informaciju o broju likova, nakon čega svaki od narednih redova sadrži podatke o po jednom liku. Ovi podaci su predstavljeni u sljedećem obliku:

κr – Krug radijusa r (npr. $\kappa 5$)
 $P a, b$ – Pravougaonik sa stranicama a i b (npr. $P 3, 2$)
 $T a, b, c$ – Trougao (trokut) sa stranicama a, b i c (npr. $T 3, 4, 5$)

Radi jednostavnosti, pretpostavite da datoteka ne sadrži neispravne podatke. Zadatak možete uraditi bez obzira da li ste uradili predhodne zadatke – dovoljno je samo da pretpostavite da klase sa navedenim svojstvima *postoje*.

II PARCIJALNI ISPIT IZ PREDMETA “TEHNIKE PROGRAMIRANJA” (GRUPA B)

VAŽNA NAPOMENA:

U svim klasama koje treba razviti, trivijalne metode koje se mogu implementirati u jednoj ili najviše dvije naredbe možete implementirati direktno unutar deklaracije klase, dok sve ostale metode trebete implementirati izvan deklaracije klase. Također, sve metode koje su po svojoj prirodi inspektori, obavezno deklarirajte kao takve.

Zadatak 1 (7 poena):

Razvijte apstraktnu baznu klasu koja predstavlja neki geometrijski lik. Jedini atributi (privatni) koje ova klasa treba sadržavati su naziv lika (tipa “string”) i interni cjelobrojni brojač, čija će uloga biti objašnjena kasnije. Interfejs klase treba da sadrži sljedeće elemente:

- Konstruktor sa jednim parametrom, koji postavlja naziv lika na vrijednost zadanu parametrom i inicijalizira pomenuti brojač na jedinicu.
- Trivijalne metode bez parametra, koje vraćaju naziv lika odnosno vrijednost brojača pohranjenog u klasi.
- Čisto virtualnu metodu bez parametara koja je namijenjena za računanje površine lika (ova metoda će biti neimplementirana, dok će konkretne implementacije biti izvedene u klasama naslijeđenim iz ove bazne klase).
- Preklopljeni operator “<<” za ispis podataka o liku na izlazni tok, koji ispisuje podatke o liku na sljedeći način (računanje površine izvodi se pozivom odgovarajuće virtualne metode):

Lik: <naziv_lik> *Površina:* <iznos_površine>

- Preklopljene relacione operatore “<” i “>” koji vrše poredenje dva lika, pri čemu se smatra da je manji onaj lik koji ima manju površinu.
- Preklopljeni unarni operator “++” koji povećava vrijednost internog brojača pohranjenog u liku za jedinicu. Potrebno je podržati samo prefiksnu verziju ovog operatora.

Iz ove bazne klase treba naslijediti tri nove klase koje predstavljaju kvadrat, elipsu i trougao (trokut). Ove klase sadrže dodatne attribute koji specificiraju ove klase: dužinu stranice za kvadrat, dužine dvaju poluososa za elipsu, te dužine tri stranice za trougao (trokut). Svi ovi atributi su realni brojevi. Interfejs ovih klasa sadrži sljedeće izmjene u odnosu na baznu klasu:

- Konstruktori ovih klasa imaju respektivno jedan, dva odnosno tri parametra, koji služe za inicijalizaciju odgovarajućih atributa. Pored toga, svi ovi konstruktori treba da inicijaliziraju i atribut koji čuva naziv lika (naslijeđen iz bazne klase) na odgovarajući sadržaj.
- Metode za računanje površina u ovim klasama izvedene su u skladu sa tim kako se zaista računa površina za svaki od specifičnih likova (površina elipse sa poluosama a i b iznosi πab , dok površina trougla sa stranicama a , b i c iznosi $\sqrt{s(s-a)(s-b)(s-c)}$ gdje je $s = (a+b+c)/2$).

Sve ostalo u ovim klasama nasljeđuje se iz odgovarajuće bazne klase.

Zadatak 2 (10 poena):

Razvijte kontejnersku klasu koja čuva kolekciju podataka o geometrijskim likovima. Pri tome, pretpostavite da se podaci o pojedinim likovima čuvaju u primjercima klasa razvijenih u prethodnom zadatku (ovaj zadatak možete uraditi bez obzira da li ste uradili prethodni – dovoljno je samo da pretpostavite da klase sa navedenim svojstvima *postoje*). Kontejnerska klasa treba da sadrži pokazivač na prvi element dinamički alociranog niza pokazivača koji pokazuju na dinamički alocirane objekte koji sadrže stvarne podatke o pohranjenim likovima. Ova klasa još sadrži informaciju o broju pohranjenih likova, zatim informaciju o kapacitetu, odnosno maksimalnom broju likova koji se mogu



pohraniti (ova informacija se čuva u konstantnom atributu), i konačno, pokazivač na brojač pristupa klasi (on će se koristiti u konstruktoru kopije i destrukturu). Inače, svi atributi klase moraju biti u privatnoj sekciji klase. Interfejs klase treba da sadrži sljedeće elemente:

- a) Konstruktor sa jednim cjelobrojnim parametrom, koji predstavlja maksimalni broj likova koji se mogu pohraniti. Konstruktor treba da inicijalizira odgovarajuće attribute i da izvrši neophodnu alokaciju memorije. Pri tome je potrebno zabraniti da se ovaj konstruktor koristi za automatsku konverziju cjelobrojnog tipa u tip klase.
- b) Konstruktor kopije, koji prosto kopira sve attribute i povećava brojač pristupa klasi za jedinicu (tako da ćemo raditi sa plitkim kopijama). Međusobno dodjeljivanje primjeraka ove klase treba zabraniti.
- c) Destruktor, koji umanjuje brojač pristupa klasi za jedinicu, i ukoliko je on dostigao nulu, oslobađa svu memoriju koja je zauzeta tokom života primjeraka klase.
- d) Metode za dodavanje novog kvadrata, elipse i trougla (trokuta) u kolekciju. Ove metode imaju onoliko parametara koliko je potrebno da se u potpunosti opiše razmatrani lik. Pored normalnih aktivnosti, sve ove tri metode treba da provjere da li među postojećim likovima već postoje likovi sa istom površinom kao lik koji se upravo dodaje, i ukoliko postoje, da poveća njihove interne brojače. Također, sve tri metode treba da bace izuzetak u slučaju da je kolekcija popunjena. Napomena: pošto u sve tri metode ima mnogo zajedničkih stvari, poželjno je definirati pomoćnu privatnu metodu koja radi sve zajedničke stvari, da izbjegnute dupliranje koda.
- e) Metodu koja vraća ukupan broj likova u kolekciji.
- f) Metodu koja vraća referencu na lik koji ima najmanju površinu (referenca se vraća sa ciljem da bi se mogao koristiti polimorfizam).
- g) Metodu koja vraća koliko ima kvadrata pohranjenih u kolekciji (preporučljivo je koristiti funkciju "count_if" iz biblioteke "algorithm", uz definiranje odgovarajuće funkcije kriterija).
- h) Metodu koja sortira sve likove pohranjene u kolekciji u rastući poredak po iznosu površine. Za tu svrhu, koristiti funkciju "sort" iz biblioteke "algorithm", uz definiranje odgovarajuće funkcije kriterija.
- i) Preklopljeni unarni operator "!", koji vraća logičku vrijednost "true" ako je kolekcija prazna, inače vraća "false".
- j) Preklopljeni operator "<<" za ispis podataka o kolekciji na izlazni tok, koji ispisuje podatke o svim pohranjenim likovima (svaki lik u posebnom redu).
- j) Preklopljen operator indeksiranja "[]" koji omogućava da se pristupi i -tom pohranjenom liku, gdje je indeks i naveden kao parametar u uglastim zagradama. U slučaju da je indeks izvan opsega, treba baciti izuzetak. Ovaj operator treba da vrati konstantnu referencu na pohranjeni lik.

Zadatak 3 (3 poena):

Napišite glavni program koji će iz tekstualne datoteke "LIKOVI_B.TXT" pročitati podatke u likovima, upisati ih u kolekciju, sortirati u rastući poredak po iznosu površine i ispisati podatke o svim likovima o kolekciji na ekran. Datoteka je organizirana tako da prvi red sadrži informaciju o broju likova, nakon čega svaki od narednih redova sadrži podatke o po jednom liku. Ovi podaci su predstavljeni u sljedećem obliku:

Ka – Kvadrat sa stranicom a (npr. K5)
 Ea, b – Elipsa sa stranicama a i b (npr. E3, 2)
 Ta, b, c – Trougao (trokut) sa stranicama a, b i c (npr. T3, 4, 5)

Radi jednostavnosti, pretpostavite da datoteka ne sadrži neispravne podatke. Zadatak možete uraditi bez obzira da li ste uradili predhodne zadatke – dovoljno je samo da pretpostavite da klase sa navedenim svojstvima *postoje*.