3 Nasleđivanje

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 destruktor);
- 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
- Broj zadatka koji radite se odredjuje na isti način kao u prvoj vežbi

Zadatak 0: Na programskom jeziku C++ implementirati klasu Funkcija ($f(x) = kx + n$). Klasa sadrži tri zaštićena atributa: koeficijente *k* i *n* (predstavljene realnim brojevima) i naziv funkcije (znakovni niz zapamćen u dinamičkoj zoni memorije). Takođe, poseduje i sledeće javne metode:

- podrazumevani konstruktor koji inicijalizuje koeficijente *k* na 1, *n* na 0, a naziv na „Linerna funkcija“,
- konstruktor kojim se postavljaju vrednosti *k* i *n*,
- virtualnu metodu koja određuje vrednosti funkcije za *x*,
- virtualni destruktor koji briše podatke iz dinamičke zone memorije ukoliko postoje,
- virtualnu funkciju *prikaziFunkciju* za štampanje atributa klase

Iz klase Funkcija izvesti klase KvadratnaFunkcija, oblika $f(x) = (x + n)^k$, i EksponencijalnaFunkcija, oblika ($f(x) = k^{x+n}\ gde\ je\ k = e$. Obe klase treba da implementiraju sledeće metode:

- podrazumevani konstruktor,
- konstruktor kojim se postavljaju vrednosti svih atributa,
- predefinisanu funkciju *prikaziFunkciju* koja štampa podatke o klasi,
- destruktor koji briše podatke iz dinamičke zone memorije ukoliko postoje.

U funkciji main kreirati niz od 2019 pokazivača na objekte klase Funkcija. Setovati niz tako da sadrži objekte sve tri klase po 673 puta. Prikazati podatke iz niza na standardni izlaz. Za proizvoljno unetu vrednost *x* prikazati podatke o funkciji koja ima minimalnu vrednost.

Zadatak 1: Na programskom jeziku C++ implementirati klasu Figura. Klasa sadrži dva zaštićena atributa: broj stranica i niz stranica (niz realnih brojeva zapamćenih u dinamičkoj zoni memorije). Takođe, poseduje i sledeće javne metode:

- podrazumevani konstruktor koji inicijalizuje broj stranica na 0,
- konstruktor kojim se postavljaju stranice kao i njihov broj,
- virtualnu metodu koja računa površinu,
- virtualni destruktor koji briše podatke iz dinamičke zone memorije ukoliko postoje,
- virtualnu funkciju *Prikazi* za štampanje atributa klase.

Iz klase Figura izvesti klase Kvadrat i Pravougaonik. Obe klase treba da implementiraju sledeće metode:

- podrazumevani konstruktor,
- predefinisanu funkciju *Prikazi* koja štampa podatke o klasi,
- destruktor koji briše podatke iz dinamičke zone memorije ukoliko postoje.

Za klasu Kvadrat dodati konstruktor kojim se sve stranice setuju na vrednost *a*.
Za klasu Pravougaonik dodati konstruktor kojim se sve stranice setuju odgovarajućim vrednostima *a* i *b*.

U funkciji main kreirati niz od 2018 pokazivača na objekte klase Figura. Setovati niz tako da sadrži po najmanje 1009 objekata klasa Kvadrat i Pravougaonik. Prikazati podatke iz niza na standardni izlaz. Sortirati niz po površini, a zatim sačuvati niz u tekstualni fajl.

Zadatak 2. Na programskom jeziku C++ implementirati klasu Point. Klasa sadrži dva zaštićena atributa: koodrinate x i y (predstavljene realnim brojevima) i sledeće javne funkcije:

- podrazumevani konstruktor koji inicijalizuje vrednosti koordinata na 0,
- konstruktor kojim se postavljaju vrednosti x i y koordinata,
- funkciju za određivanje rastojanja između dve tačke,
- virtualni destruktor,
- virtualnu funkciju *printData* za štampanje atributa klase.

Iz klase Point javno izvesti klasu PointCity koja označava lokaciju grada na geografskoj karti. Ova klasa sadrži privatne atribute - naziv grada (znakovni niz zapamćen u dinamičkoj zoni memorije), naziv države kojoj pripada (znakovni niz zapamćen u dinamičkoj zoni memorije) i broj stanovnika, kao i sledeće javne funkcije:

- podrazumevani konstruktor,
- konstruktor kojim se postavljaju vrednosti svih atributa,
- predefinisanu funkciju *printData* koja štampa podatake o klasi (dodajte naziv i broj stanovnika),
- destruktor koji briše podatke iz dinamičke zone memorije ukoliko postoje.

U funkciji main kreirati niz objekta klase PointCity na osnovu podataka iz unapred spremljenog fajla (LV3Zad3.txt). Podaci su upisani u formatu naziv grada, pa naziv države kojoj pripada pa broj stanovnika. Podaci su međusobno odvojeni tabulatorom, a nazivi gradova i država mogu da imaju

blanko znak u sebi, pa je potrebno da i o tome vodite računa.
Npr:
Belgrade Serbia 1,166,763[210]
Los Angeles United States 3,884,307[103]

Obratite pažnju da broj stanovnika nije prikazan kao standardni integer, nego da ima i simbole grupisanja koje bi trebalo da izbacite. Takođe, nakon broja stanovnika (u većini slučajeva) stoji i referenca(ili reference) na izvor koje treba da uklonite pre konvertovanja tekstualne u brojnu vrednost. U datom primeru 1,166,763[210] treba da se evaluira kao 1166270.
Listu gradova sortirati po broju stanovnika i sortiranu listu upisati u fajl na isti način kao što je lista iz početnog fajla (podaci o jednom gradu smešteni u okviru jednog reda, najpre naziv grada, pa države i na kraju broj stanovnika).

Zadatak 3. Na programskom jeziku C++ kreirati klasu Buffer koja kao privatne članove sadrži:

- kapaciteti bafera,
- broj upisanih elemenata u bafer,
- dinamički niz elemenata tipa integer koji predstavlja sam bafer.

Klasa sadrži i javne članove:

- konstruktor koji inicijalizuje kapacitet bafera,
- destruktor,
- virtuelnu funkciju *push* za dodavanje novog elementa na kraj bafera,
- virtuelnu funkciju *pop* za izbacivanje elementa sa početka bafera.
- Virtuelnu funkciju Clear koja izbacuje sve elemente iz bafera

Iz klase Buffer izvesti klase QueueBuffer i OrderedBuffer.
Funkcionalnosti za QueueBuffer treba da budu sledeće:

- Funkcija push treba da doda element na prvo slobodno mesto u bafer.
- Funkcija Clear funkcioniše kao i ona iz osnovne klase.
- Funkcija *pop* treba da izbacuje pretposlednji element iz bafera. Ako bafer ima samo jedan element, onda treba da izbaci njega.

Funkcionalnosti za OrderedBuffer treba da budu sledeće:

- funkcija Clear treba da vrednosti svih elemenata postavi na 0.
- Funkcija push treba da doda novi elemenat u bafer tako da sadržaj bafera ostane sortiran u rastućem redosledu.
- Funkcija pop treba da izbaci elemenat iz sredine bafera.

U funkciji main deklarisati tri pokazivača na klasu Buffer, a zatim dinamički kreirati objekte klase Buffer, QueueBuffer i PostponedStackBuffer i testirati sve funkcije članice, tako što ćete kroz 2018 puta da ponovite sledeću sekvencu za svaki od objekata:

- Tri puta pozovete push sa slučajno generisanim celi brojem
- Dva puta pozovete pop
- U svakom pedesetom ciklusu zovete Clear
- Nakon svakog stotog ciklusa štampate sadržaj vaših bafera

Zadatak 4. Na programskom jeziku C++ kreirati klasu Command koja predstavlja apstrakciju komandi (npr. “Print”, “Save”, “Open”, “Draw”) i koja kao članove ima:

- *title* - naziv komande,
- *value* – vrednost parametra (char*),
- *konstruktor koji postavlja parametre title i value,*
- virtualnu funkciju *execute()* (izvršenje komande) koja ispisuje naziv komande (title).

Iz klase Command izvesti tri klase: Command Draw, Command Save, Command Print. Svaka klasa definiše *title* (Command Draw -> “Draw”, Command Save -> “Save“, Command Print -> “Print“) i predefiniše execute funkciju (ispisuje parametre title i value za izvršene operacije).
Kreirati klasu CommandHistory koja pamti komande i ima realizovan Undo mehanizam tako što klasa sadrži sledeće članove:

- vektor pokazivača na objekte klase Command veličine 2019,
- trenutni broj zapamćenih komandi,
- funkciju undo() koja briše zadnji element liste,
- funkciju create(int x) koja na osnovu parametra x kreira odgovarajuću komandu (objekat klase Command Draw, Command Save ili Command Print - ukoliko u vektoru komandi nema više mesta briše se prva upisana komanda i dodaje nova),
- funkciju execute() koja poziva method execute() zadnje izdate komande.

U funkciji main kreirati objekat klase CommandHistory, i testirati sve funkcije svih kreiranih klasa. Napuniti CommandHistory sa po 673 Draw, Save i Print komandi izvršiti ih i isprazniti bafer.

Zadatak 5. Na programskom jeziku C++ definisati:

- Klasu Broj koja sadrži:
  - Atribut *vrsta* (char*),
  - Atribut *vrednost* (double),
  - javnu funkciju za poređenje dva broja (po vrednosti),
  - virtuelnu javnu metodu Print za prikaz vrste i vrednosti broja na standardni izlaz,
  - zaštićenu virtuelnu metodu za postavljanje vrednosti broja,
  - virtuelnu metodu za vraćanje vrednosti broja.
- Klasu RacionalanBroj izvedenu iz klase Broj (za predstavljanje brojeva oblika a/b, gde su a i b celi brojevi).
- Klasu KompleksanBroj izvedenu iz klase Broj za predstavljanje brojeva a+*j*b. Pod vrednošću broja u ovom slučaju podrazumeva se njegov modul.

U izvedenim klasama definisati privatne atribute kojima je određen odgovarajući broj i konstruktore koji postavljaju odgovarajuće atribute.
U funkciji main, kreirati po niz od 2018 pointera na objekte tipa broj, od kojih 1009 ukazuju na racionalne a 1009 na kompleksne brojeve. Vrednosti brojeva ili učitati iz unapred definisanog fajla, ili ih slučajno generisati. Sortirati brojeve u nizu po vrednosti u opadajućem redosledu, i upisati ih u tekstualni fajl.

Zadatak 6. Na programskom jeziku C++ kreirati klasu Window koja kao članove ima:

- *title* - naziv prozora,
- *state* – stanje prozora (otvoren/zatvoren),
- virtualnu funkciju *draw()* koja ispisuje vrednosti parametra title i state,
- funkciju *open*() – otvaranje prozora,
- funkciju *close*() – zatvaranje prozora.

Pri otvaranju i zatvaranju prozora se modifikuje stanje prozora (state). Iz klase Window izvesti dve klase: DialogWindow, DocumentWindow. Klasa DialogWindow predefiniše funkciju draw tako što ispisuje na ekranu "DialogWindow nacrtan", i ima dodatne dve funkcije:

- *confirm*() – zatvara prozor i zatim vraća vrednost 1,
- *cancel*() – zatvara prozor i vraća vrednost 0.

Klasa DocumentWindow predefiniše draw tako što ispisuje na ekranu "DocumentWindow nacrtan".

U funkciji main definisati niz od 2018 pokazivača tipa Window, kreirati po 1009 objekata klase DialogWindow i DocumentWindow i testirati sve funkcije kreiranih klasa. Na kraju sortirati sve prozore po nazivu i sortirani niz upisati u tekstualni fajl.

Zadatak 7. Na programskom jeziku C++ kreirati klasu Artikal koja kao članove ima:

- naziv,
- cenu,
- virtualnu funkciju showDescription() koja ispisuje parametre naziv i cena,
- funkciju getPrice() koja vraća cenu artikla.

Iz klase Artikal naslediti dve klase: Laptop i Torba. Klasa Laptop poseduje sledeće članove:

- opis (char*),
- stanje (uključen-isključen),
- predefinisanu funkciju showDescription() koja ispisuje na ekranu naziv, cenu i opis,
- funkciju turnOn() – modifikuje se stanje i ispisuje se poruka na ekranu,
- funkciju turnOff() – modifikuje se stanje i ispisuje se poruka na ekranu.

Klasa Torba sadrži sledeće članove:

- artibut sadržaj – tipa Artikal, pokazivač na artikal trenutno smešten u torbi,
- predefinisanu funkciju showDescription() koja ispisuje na ekranu tekst "Torba za Laptop računar" i nakon toga zove metodu showDescription za artikal smešten u torbi,
- funkciju put(Artikal& a) – simulacija smeštanja artikla u torbu ukoliko je prazna i izpisuje poruku da li je artikal smešten ili ne,
- funkciju remove() – simulira vađenje artikla iz torbe.

U funkciji main kreirati niz od 2018 pointera tipa Artikal i po 1009 objekata klase Laptop i Torba i testirati sve funkcije obe klase. Na kraju, sortirati sve proizvode po ceni i rezultat sortiranja upisati u tekstualni fajl.

Zadatak 8. Na programskom jeziku C++ kreirati klasu GeometrijskaSlika koja sadrži privatne atribute:

- boja (definisana svojim komponentama: R-crveno, G-zeleno i B-plavo čije su vrednosti u opsegu 0-255),
- težište (određeno x i y koordinatama).

Klasa sadrži sledeće javne metode:

- konstruktor koji inicijalizuje sve privatne atribute,
- metodu za izračunavanje rastojanja figure (njenog težišta) od koordinatnog početka,
- metodu za transliranje figure za zadati pomeraj (definisan svojim x i y komponentama),
- virtuelnu metodu Show za prikaz atributa boja i težište na standardni izlaz,
- zaštićenu virtuelnu metodu za izračunavanje površine figure,
- metodu za ispitivanje da li je figura veća (po površini) od druge zadate figure.

Kreirati i klase Krug (čiji je privatni atribut poluprečnik) i Kvadrat (čiji je privatni atribut stranica). U izvedenim klasama definisati konstruktore, funkcije za izračunavanje površine. Predefinisati funkcije za prikaz atributa tako što će pre prikaza atributa biti ispisan tip figure, a posle prikaza atributa roditeljske klase prikazati i atribute definisne u izvedenim klasama.
U funkciji main kreirati po 2018 objekata klase Kvadrat i Krug u dinamičkoj zoni memorije. Postaviti sve njihove atribute na slučajne vrednosti iz opsega 0-255. Nakon toga translirati ih za vektor (a, b). Za svaku figuru odrediti posebno a i b kao slučajno izabranu vrednost iz intervala (-12.8, 12.8).
Sortirati sve kreirane figure po površini u neopadajućem redosledu i u tekstualni fajl upisati sve njihove atribute, površinu i rastojanje od koordinatnog početka. Voditi računa da se podaci o jednoj figuru upisuju u jedan red izlaznog fajla.

Zadatak 9. Na programskom jeziku C++ kreirati:

- Klasu Displej koja sadrži:
    - zaštićeni podatak *cifra* koja predstavlja hex cifru trenutno prikazanu na displeju,
    - virtuelnu funkciju *set* koja postavlja vrednost cifre. Ova funkcija treba da obezbedi da se nevalidne vrednosti ne prihvate
    - funkciju *reset* koja postavlja cifru na 0,
    - virtuelnu funkciju *increment* koja povećava sadržaj displeja za 1 po modulu 16,
    - virtuelnu funkciju *show* koja upisuje sadržaj displeja na standardni izlaz.
- Klasu DekadniMatricniDisplej javno izvedenu iz klase Displej koja sadrži:
    - privatni podatak - matricu piksela reda mxn u kojoj su nulama i jedinicama predstavljene ugašene i upaljene tačke na displeju koje formiraju cifru.
    - U klasi DekadniMatricniDisplej dodati funkciju koja će unapred definisane sadržaja matrice u skladu sa cifrom koja je trenutno prikazana na displeju, učitati iz fajla.
    - Predefinisati funkciju set, da prihvata samo dekadne cifre
    - Predefinisati funkciju inkrement taka da sadržaj displeja povećava za 1 po modulu 10.

- Funkcija show treba da najpre pozove na izvršenje metodu show iz osnovne klase, a zatim ispiše sadržaj matrice piksela na standardni izlaz.

U funkciji main kreirati po jedan objekat klasa Displej i DekadniMatricniDisplej.
Za oba objekta izvršiti sledeću sekvencu akcija:

- Postaviti inicijalnu vrednost na 9
- Izvršiti k inkrementiranja (k mora da bude veće od 50000)
- Pozvati funkciju reset
- Izvršiti k inkrementiranja (k mora da bude veće od 50000)

Nakon poslednjeg, kao i nakon svakog 2018.og inkrementiranja, pozvati metodu show za svaki od objekata.