***Vaje C++ - preoblaganje konstruktorjev, preoblaganje operatorjev, šablone, kazalci na razrede***

1. Deklarirajte razred *Pravokotnik*, ki je izpeljan iz razreda *Kvadrat*. Razred *Kvadrat* naj vsebuje samo lastnost stranica\_a, privzeti konstruktor ter destruktor. Razred *Pravokotnik* naj še dodatno vsebuje lastnost stranica\_b, metodo za vnos lastnosti, metodo za izpis obsega in ploščine. (Formule: obseg pravokotnika = 2 (a + b), ploščina pravokotnika =a \* b). Napišite program v C++, ki bo deklariral objekt razreda *Pravokotnik*, prebral njegove lastnosti ter izpisal njegov obseg in ploščino. V glavnem programu še dinamično deklarirajte kazalec na razred Pravokotnik ter preko kazalca ponovno kličite vse metode razreda Pravokotnik. Na koncu še sprostite kazalec iz pomnilnika.
2. Napišite program v C++, ki deklarira razred *Kvader* z lastnostmi a, b in c, ki naj bodo zasebne (*private*) ter tipa celih števil. Razred naj vsebuje še metode za branje vseh lastnosti razreda, metodo za izračun prostornine kvadra (v = a \* b \* c), privzeti ter kopirni konstruktor, preobloženi operator seštevanja (x + y) ter preobloženi operator +=. Operator x + y naj sešteje prostornini kvadrov x in y. Operator += naj vsaki stranici kvadra (a, b in c) prišteje vrednost vhodnega parametra. V glavnem programu ustvarite objekt x s pomočjo privzetega konstruktorja ter objekt y s pomočjo kopirnega konstruktorja. Nato omogočite vnos lastnosti objektov x in y. Nato prištejte objektu x vrednost, ki se naj bere s tipkovnice ter seštejte objekta x in y ter izpišite prostornino tako dobljenega objekta (kvadra).

Razred spremenite tako, da bo ta uporabljal generični tip T za stranice a, b in c, ter v glavnem programu določite, da naj bo tip T tipa long in nato še tipa float.

1. Napišite program v C++, ki deklarira razred *Tocka* z lastnostmi x, y in oblika. Lastnost oblika naj bo znakovnega tipa, lastnosti x in y pa naj bosta kratki celi števili. Vse lastnosti naj bodo zasebne (*private*). Razred naj vsebuje še metodo za branje obeh lastnosti, ki ne sme dopustiti vnos negativnih vrednosti, privzeti ter pretvorbeni konstruktor, preobloženi operator inkrementa (++), preobloženi operator \*= ter preobloženi operator enačaj (a==b). Ta operator primerja med seboj dva objekta. Objekta sta enaka, kadar sta obe njuni lastnosti x in y enaki. Operator inkrementa (++) poveča lastnost x za 1, operator \*= pa naj množi lastnosti x in y za toliko, kolikor znaša vrednost vhodnega parametra. V glavnem programu ustvarite objekt a s pomočjo privzetega konstruktorja ter objekt b s pomočjo pretvorbenega konstruktorja (pretvorite v tip float). V glavnem programu omogočite vnos lastnosti obeh objektov a in b. Za objekt a kličite operator ++, za objekt b pa operator b+=3. Če sta objekta enaka, naj program izpiše ustrezno sporočilo, če pa ne, pa naj program izpiše kakšna je razdalja med njima (d=sqrt(pow(x1-x0,2)+pow(y1-y0,2));). Za računanje razdalje uporabite vmesnike (interface). V glavnem programu še dinamično deklarirajte kazalec na razred Tocka ter preko kazalca ponovno kličite vse metode razreda Tocka. Ali pride pri klicanju preobloženih operatorjev s pomočjo kazala do kakšnih napak? Zakaj? Na koncu še sprostite kazalec iz pomnilnika.

Razred še spremenite tako, da bo ta uporabljal generični tip T za koordinati x in y ter v glavnem programu določite, da naj bo tip T tipa short in nato še tipa long.

1. Napišite program v C++, ki deklarira razred *Akvarij* s privatno lastnostjo ribe, ki naj vsebuje največ 100 realnih števil, ki predstavljajo dolžine rib (v cm), ki jih naselimo v akvarij. V razredu naj bo tudi celoštevilčna lastnost vol, ki predstavlja prostornino akvarija (v l) . Razred naj vsebuje privzeti (vse dolžine-rib v tabeli artikli postavi na 0), ter destruktor. Razred naj vsebuje metodo za izpis lastnosti ribe in in vol, metodo za branje, ki bere dolžine rib s tipkovnice in jih shranjuje v tabelo ribe, dokler pravilo o zasedenosti akvarija ni preseženo. Ta pravilo pa pravi, da za vsak cm ribe potrebujemo vsaj 2 litra vode. Pri tem metoda za vnos ne sme dovoliti vnosa negativnih vrednosti dolžin rib. Razred naj vsebuje tudi metodi za izračun zasedenosti akvarija (v %) in za izračun povprečno dolžine vseh rib v akvariju. Obe metodi naj vračata vrednosti. Razred naj vsebuje še preobložene operatorje == , -=, += . Operator == naj vrne vrednost 1, če sta prostornini dveh akvarijev enaki in sta seznama rib enaka. Pri tem ni nujno, da so velikosti rib vnesene v enakem vrstnem redu. Operator -= naj odstrani tisto ribo iz seznama, ki se nahaja na n-tem mestu v lastnosti ribe, pri čemer naj bo n vhodni podatek tega operatorja. Operator naj deluje tako, da prestavi vse elemente tabele ribe za eno mesto levo od mesta brisanja naprej. Operator += naj omogoča dodajanje nove ribe v akvarij (vhodni podatek n predstavlja dolžino ribe), a le če s tem nismo presegli pravilo o naselitvi akvarija.

V glavnem programu ustvarite objekta morski in sladkovodni s pomočjo privzetega konstruktorja ter kliči metodi za vnos in izpis za oba objekta. Za objekt morski dvakrat kliči operator += (pred tem naj se vrednosti vhodnih podatkov berejo s tipkovnice) ter operator -=. Nato ponovno kliči metodo za izpis ter metodo za izračun zasedenosti akvarija za objekt morski. Na koncu še izpiši rezultata primerjave med objektoma morski in sladkovodni. Pred zaključkom programa sprosti oba objekta iz pomnilnika.

Program spremenite tako, da bo ta uporabljal razredne šablone (templates). Tabela artikli naj bo generičnega podatkovnega tipa, katerega bomo določili šele v glavnem programu (izberi si dva različna tipa po lastni izbiri).

1. Napišite program v C++, ki deklarira razred Tabela. Razred naj vsebuje privatno lastnost tab, ki naj bo tabela največ 100-ih elementov generičnega tipa T. Razred naj vsebuje še privatno lastnost koliko, ki vsebuje vrednost dejanskega števila elementov tabele (ni namreč nujno, da uporabnik vnese vseh 100 elementov). V razredu naj bodo še naslednje javno dostopne metode: vnos, izpis, operator +, operator > in operator -=.

Metoda za vnos naj deluje tako, da lahko uporabnik prekine vnos elementov tabele že prej, torej ni nujno, da vnese vseh 100 elementov tabele in poskrbi zato, da koliko vsebuje vrednost dejanskega števila elementov tabele. Operator plus naj vrne vsoto (ki je zopet objekt razreda Tabela) dveh objektov razreda Tabele tako, da na konec tabele 1. objekta doda vse elemente tabele 2. objekta. V kolikor to ni možno izvesti (vsota elementov bi presegla število 100), potem naj operator + vrne napako. Operator > vrne logično vrednost 1, v kolikor je vsota elementov tabele 1. objekta večja od vsote elementov tabele 2. objekta. V nasprotnem primeru operator > vrne 0. Operator += naj vsem elementom tabele odšteje vrednost n (parameter operatorja).

V glavnem programu deklarirajte dva objekta razreda Tabela, njun tip naj bo definiran kot celo število in prikažite klic vseh metod razreda Tabela. Nato še deklarirajte en objekt razreda Tabela, njegov tip naj bo realno število. Ali vsi operatorji delujejo? Kaj pa se zgodi pri seštevanju objektov, katerih tip T se v glavnem programu razlikuje? Zakaj?

1. Napišite program, ki deklarira Razred DLL (Dynamic link list) in po vzoru 5. naloge implementira iste metode (vnos, izpis, operator +, operator -= in operator >), ki pa naj ne delujejo za 1d polje, temveč za enosmerno povezani neurejeni kazalčni seznam. Pri tem naj velja, da je podatkovna lastnost seznama (data) enka generičnemu tipu T. Metoda za vnos naj kliče funkcijo push(), operator > vrne logično vrednost 1 v kolikor je vsota vseh elementov 1. seznama večja od vsote vseh elementov 2. seznama. Operator + sešteje (združi) oba seznama tako, da se konec 1. seznama dodajo vsi elementi 2. seznama (tukaj ne bo težav z omejitvijo velikost kot 1. nalogi). Razredu dodajte še destruktor, ki ob končanju programa sprosti vse elemente kazalčnega(ih) seznama(ov) iz pomnilnika (ta lahko kliče funkcijo pop).

V glavnem programu deklarirajte dva objekta razreda DLL, njun tip naj bo definiran kot celo število in prikažite klic vseh metod razreda DLL. Nato še deklarirajte en objekt razreda DLL, njegov tip naj bo realno število. Ali vsi operatorji delujejo? Kaj pa se zgodi pri seštevanju objektov, katerih tip T se v glavnem programu razlikuje? Zakaj? Kako bi seznam, implementiran v razredu DLL razvrstili po velikosti? Razmislite in dodajte metodo razreda DLL, za razvrščanje podatkov seznama v padajočem vrstnem redu.