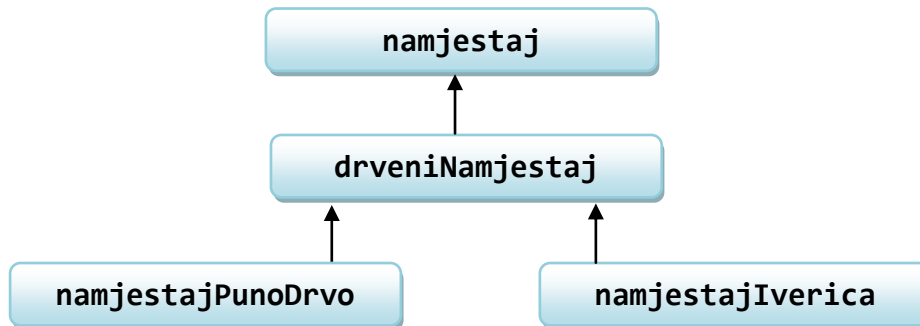


Zadatak:

Napišite sučelje i implementaciju za klase `namjestaj`, `drveniNamjestaj`, `namjestajIverica` i `namjestajPunoDrvo`. Sljedeći dijagram opisuje relacije među klasama. Sučelja za klase spremite u datoteku `namjestaj.h`, a implementacije u datoteku `namjestaj.cpp`.



Klasa `namjestaj`

Ova klasa opisuje namještaj. Svaki namještaj ima tri parametra koji ga određuju: masa, cijena i materijal.

- **`namjestaj(int masa, int cijena, string materijal)`**
Konstruktor za klasu `namjestaj`.
- **`int masa()`**
Vraća masu namještaja.
- **`int cijena()`**
Vraća cijenu namještaja.
- **`string materijal()`**
Vraća vrstu materijala od kojeg je načinjen namještaj.
- **`void spustiCijenu(int x)`**
Spušta cijenu namještaja za x kuna.
- **`int vrijednostMaterijala()`**
Vraća vrijednost materijala od kojeg je napravljen namještaj. Prava vrijednost je 10kn/kg.
- **`static namjestaj* stvoriGarnituru(namjestaj *pojedinačniElementi[], int size)`**
Stvara novi objekt tipa `namjestaj` i vraća pokazivač na njega. Masa (cijena) je suma masa (cijena) svih elemenata u polju `pojedinačniElementi`. Materijal je jednak materijalu najtežeg elementa u polju.
- **`static namjestaj& nadjiNajjeftiniji(string materijal)`**
Vraća referencu na najjeftiniji namještaj danog materijala (s obzirom na sav namještaj koji trenutno postoji u programu). Ukoliko ima više rješenja, vraća se proizvoljno rješenje.

Klasa `drveniNamjestaj`

Ova klasa opisuje drveni namještaj. Materijal je tipa "drvo". Svaki drveni namještaj ima još dva parametra koji ga opisuju: tip drva i debljina drva.

- **`drveniNamjestaj(int masa, int cijena, string tipDrva, int debljinaDrva);`**
Konstruktor za klasu `drveniNamjestaj`.
- **`int debljinaDrva()`**
Vraća debljinu drva.
- **`string tipDrva()`**
Vraća tip drva.

Klasa namjestajIverica

Ova klasa opisuje namještaj od iverice. Tip drva je "iverica".

- **namjestajIverica(int masa, int cijena, int debljinaDrva)**
Konstruktor za klasu namjestajIverica.
- **vrijednostMaterijala()**
Vraća vrijednost materijala od kojeg je napravljen namještaj. Prava vrijednost je 5kn/kg.

Klasa namjestajPunoDrvo

Ova klasa opisuje namještaj rađen od punog drveta. Tip drva je "puno_drvo".

- **namjestajPunoDrvo(int masa, int cijena, int debljinaDrva)**
Konstruktor za klasu namjestajPunoDrvo.
- **int vrijednostMaterijala()**
Vraća vrijednost materijala od kojeg je napravljen namještaj. Prava vrijednost je 20kn/kg.
- **namjestajIverica* jeftinaKopija()**
Vraća pokazivač na objekt tipa namjestajIverica čija je masa (cijena) jednaka polovini (trećini) mase (cijene) objekta koji je pozvao funkciju. Debljina drva ostaje ista. Masa (cijena) će uvijek biti djeljivi s 2 (3).

Primjer klijentskog programa

```
#include "namjestaj.h"
#include <iostream>

using namespace std;

int main(void)
{
    namjestaj A(10, 2300, "drvo"), B(20, 3000, "aluminij"), C(5, 1200, "plastika");
    namjestaj *nizNamjestaja[5];
    nizNamjestaja[0] = new drveniNamjestaj(30, 1500, "hrast", 4);
    nizNamjestaja[1] = new namjestajIverica(10, 1000, 2);
    nizNamjestaja[2] = new namjestajPunoDrvo(25, 4000, 5);
    nizNamjestaja[3] = new namjestajPunoDrvo(20, 3000, 3);
    nizNamjestaja[4] = new namjestajIverica(5, 1100, 1);

    namjestaj &jeftinoDrvo = namjestaj::nadjinaJeftiniji("drvo");
    cout << jeftinoDrvo.masa() << " " << jeftinoDrvo.cijena() << " " << jeftinoDrvo.materijal() << endl;
    // 10 1000 drvo

    A.spustiCijenu(1500);
    jeftinoDrvo = namjestaj::nadjinaJeftiniji("drvo");
    cout << jeftinoDrvo.masa() << " " << jeftinoDrvo.cijena() << " " << jeftinoDrvo.materijal() << endl;
    // 10 800 drvo

    for(int i = 0; i < 5; i++)
        cout << nizNamjestaja[i]->vrijednostMaterijala() << endl;
    // 3000
    // 500
    // 5000
    // 4000
    // 250

    namjestajPunoDrvo D(10, 1800, 3);
    namjestajIverica *jeftino = D.jeftinaKopija();
    cout << jeftino->masa() << " " << jeftino->cijena() << " " << jeftino->materijal() << " " << jeftino-
>tipDrva() << " " << jeftino->debljinaDrva() << endl;
    // 5 600 drvo iverica 3

    namjestaj *elementiZaGarnituru[3];
    elementiZaGarnituru[0] = new namjestaj(5, 500, "aluminij");
    elementiZaGarnituru[1] = new namjestajPunoDrvo(20, 3000, 4);
    elementiZaGarnituru[2] = new namjestaj(15, 1000, "plastika");
    namjestaj *garnitura = namjestaj::stvariGarnituru(elementiZaGarnituru, 3);
    cout << garnitura->masa() << " " << garnitura->cijena() << " " << garnitura->materijal() << endl;
    // 40 4500 drvo

    return 0;
}
```

Opće napomene

- Struktura, funkcije i datoteke koje šalžete moraju se zvati točno onako kako je zadano u zadatku. Pazite na mala i velika slova!
- Trebate poslati samo sučelje i implementaciju. U datotekama koje šalžete ne smije se nalaziti funkcija `main()`!
- nijedna funkcija ne smije ništa učitavati s tipkovnice ili neke datoteke, niti išta ispisivati na ekran ili u neku datoteku.
- Svaki od `main`-ova pomoću kojih testiramo ispravnost vašeg programa neće pozivati sve gore navedene funkcije. Stoga, ako neku od funkcija ne znate napisati ipak možete dobiti koji bod (u tom slučaju tu funkciju nemojte navesti niti u `.h` niti u `.cpp` datoteci ili napravite neku trivijalnu implementaciju).

Ispravnost implementacija koje napišete bit će provjerena tako da ćemo mi napisati razne klijentske programe koji će deklarirati nekoliko varijabli zadane strukture, i na njima pozivati funkcije koje ste trebali napisati. Ako se poslani programi ne budu uspješno povezivali (linkali) s našim klijentskim programima, smatrat će se neispravnima.

Neki klijentski programi provjeravat će samo neke jednostavnije funkcije, dok će neki provjeravati sve funkcije koje trebate napisati. Provjera je potpuno automatska, tako da je od presudne važnosti da se pridržavate specifikacije. Nepridržavanje lako može uzrokovati osvojenih 0 bodova iz zadaće!

Naravno, za provjeru radi li implementacija prije nego što je pošaljete, preporučuje se da je testirate pomoću nekog klijentskog programa. No taj klijentski program ne šalžete!

Za sva pitanja vezana uz ovu zadaću javite se asistentu Vujčiću na ivujcic+rp1@gmail.com.