Ülesanne „Warenhouse“.

Teie sõber peab ladu, mille haldamine on siiani toimunud paberi peal. Sõber soovib lihtsat laohaldustarkvara (warehouse management system), mis võimaldaks lattu saabuvaid tooteid registreerida ning tooteid laost üles otsida.

Toode (**Product**) on ese, millel on nimi ja suurus. Sama nime ja suurusega tooteid käsitleme samaväärsetena. Klassis **Product** looge meetod **bool equals(Product \*p),** mis tagastab väärtuse ***true***, kui toode ***this*** on samaväärne tootega p.

Lao asukohti arvestatakse riiuli lõikes, igal riiulil on tekstiline nimetus (nt. A, B, C jne). Ühel riiulil võib asuda mitu toodet, samas võib nt. „Sokid suurus 42“ asuda laos mitmel erineval riiulil. Lisaks võib ühel riiulil olla sama toodet mitu tükki.

Koostada klass **Warehouse**, mis sisaldab endas kaupade asukohtadele paigutamiseks vajalikku andmestruktuuri. Realiseerida järgmised meetodid:

* **receive(prod, loc)** – lisab lattu uue toote prod ning paigutab selle riiulile loc.
* **getItemCount(prod)** - tagastab etteantud toote prod kohta, mitu tükki seda laos leidub. Nt. „Sokid nr. 42“ on laos 2 tükki.
* **dispatch(prod)** - otsib laost etteantud toodet prod.

Kui toode on laos, tagastatakse selle asukoht ning vastava asukohaga toode eemaldatakse laost. Kui laos on sama toodet mitu tükki, väljastatakse ikkagi 1 toode (nt. esimesel kättejuhtuval asukohal paiknev). Kui laos toodet ei ole, tagastab meetod väärtuse NULL.

Loodud klasside ja meetodite testimiseks kasutage järgmist testprogrammi:

//loome kaubad

Product \*ps1 = new Product("sokid", 42);

Product \*ps2 = new Product("sokid", 42);

Product \*pk1 = new Product("kindad", 10);

Product \*pk2 = new Product("kindad", 9);

Product \*pm = new Product("mantel", 60);

Product \*pp = new Product("püksid", 56);

// Kontrollime equals() toimimist

cout<<ps1->equals(ps1)<<endl; // true

cout<<ps1->equals(ps2)<<endl; // true

cout<<ps2->equals(ps1)<<endl; // true

cout<<ps1->equals(pk1)<<endl; // false

// Testime lao toimimist

// lisame kaubad lattu

Warehouse w = new Warehouse();

w.receive(ps1, "A");

w.receive(pk1, "A");

w.receive(pk2, "B");

w.receive(ps2, "B");

w.receive(pm, "B");

w.receive(pp, "L");

cout<<ps1->to\_string()<<" on laos " << w.getItemCount(ps1) <<" tk"); //2tk

cout<<ps2->to\_string()<<" on laos " << w.getItemCount(ps2) <<" tk"); //2 tk

cout<<pk1->to\_string()<<" on laos " << w.getItemCount(pk1) <<" tk"); //1 tk

// loome otsitava kauba

Product \*po1 = new Product("sokid", 42);

// väljastame laost kauba:

if (w.dispatch(po1) != NULL) {

 cout<< "Väljastasin toote " << po1->to\_string() << " asukohalt " <<w.dispatch(po1) <<endl;

} else {

 cout<< "Toodet "<< po1->to\_string()<<" ei leitud loast"<<endl;

}

// kontrolliks prindime välja laoseisu ja vaatame, kas antud kauba kogus on vähenenud

cout<<ps1->to\_string()<<" on laos " << w.getItemCount(ps1) <<" tk"); //1tk