

Објектно оријентисано програмирање

Септембар 2024

Опште напомене:

- Атрибути не могу бити јавни.
- Написати све потребне јавне методе које ће обезбедити да програм ради (укључујући конструкторе и деструктор).
- Избећи на сваки начин дуплирање кода у целом пројекту.
- Придржавати се основних принципа објектно оријентисаног програмирања.

Задатак 1:

Написати програм који ће фабрици пића помоћи да води евиденцију о различитим паковањима воде и сокова. О води се памти запремина амбалаже у литрима и број таквих амбалажа. О соковима се, поред наведена два податка, памти и укус сока (низ карактера), проценат воћа у соку (преостали садржај сока је вода) и однос цене воде и воћа у соку.

Ново пиће се додаје у евиденцију тако што се повећа број одговарајућих амбалажа тог пића за један ако постоји барем једно такво пиће, у супротном се додаје на крај колекције.

Уколико треба да се изнесе одређено пиће из складишта, прво се проверава да ли постоји тражено пиће у одговарајућој амбалажи и након тога се смањује број амбалажа. Пиће се брише из колекције уколико број амбалажа падне на нулу. Уколико нема довољно амбалажа, приказати поруку.

Корисник у сваком тренутку може видети садржај складишта исписивањем свих података о сваком пићу.

Могуће је пресипати једно пиће задате амбалаже у одговарајући број мањих амбалажа задате запремине.

Корисник може проверити да ли има довољну количину пића да испуни захтев клијента за задатим бројем литра пића.

Могуће је вратити два пића за која важи да је укупна цена тог пића најмања, односно највећа, једним проласком кроз колекцију.

Проверити рад апликације функцијом `zadatak1` која се налази у наставку.

Задатак 2:

Написати програм о вођењу евиденције у фабрици пића. Програм треба да подржи рад са различитим типовима података који описују пиће и да обезбеди чување произвољног броја ових података.

Могуће је додавање новог пића у колекцију која представља фабрику, али је неопходно да она остане уређена у неоппадајући редослед по запремини. Уколико је колекција пуна пријавити проблем.

Могуће је избацивање пића из колекције на основу задате запремине. Ако не постоји такво пиће пријавити проблем.

Омогућити одређивање и враћање укупне запремине пића у складишту.

Омогућити и враћање укупног броја амбалажа задате запремине и укупне запремине пића у тој амбалажи, једним проласком кроз колекцију.

Омогућити одређивање и враћање најмањег броја амбалажа којим је могуће испунити захтев за траженом количином пића (запремина пића може бити већа или једнака траженој количини), ако није могуће вратити 0.

Апликација може сачувати колекцију у датотеку за каснију анализу, а могуће је и учитати запамћене податке и наставити рад са њима.

Рад апликације треба проверити за случај фабрике која производи воду и за случај фабрике сокова. Фабрика воде за сваку амбалажу памти само запремину у литрима, тачније памти само реалне податке. Фабрика сокова памти назив воћа (низ карактера), запремину амбалаже и проценат воћа.

Проверити рад апликације функцијом *zadatak2* која се налази у наставку. Проширити програм тако да обезбеди обраду изузетака.

ПРЕДМЕТНИ НАСТАВНИЦИ И АСИСТЕНТИ

main.cpp

```
void zadatak1()
{
    char ukus1[7] = "jagoda";
    int procenat1 = 50;
    float odnosCene = 0.75;

    // 2 poena
    Pice* p1 = new Voda(1);
    //p1->Prikazi();
    cout << *p1 << endl;
    // 2 poena
    Pice* p2 = new Sok(0.75, ukus1, procenat1, odnosCene);
    //p2->Prikazi();
    cout << *p2 << endl;

    float voda[] = { 1, 0.2, 2, 5, 1, 1, 2, 5 };
    float sok[] = { 1, 0.2, 1.5, 0.5, 1, 0.5, 1, 1.5 };
    Skladiste* s = new Skladiste(10);
    // 2 poena
    for (int i = 0; i < 4; i++)
    {
        s ->Dodaj(new Voda(voda[i]));
        s ->Dodaj(new Sok(sok[i], ukus1, procenat1, odnosCene));
    }
    // 2 poena
    //s->print();
}
```

```

cout << *s << endl;

// 2 poena
for (int i = 4; i < 8; i++)
{
    s->Dodaj(new Voda(voda[i]));
    s->Dodaj(new Sok(sok[i], ukus1, procenat1, odnosCene));
}
//s->print();
cout << *s << endl;

// 2 poena
Sok sok(0.5, ukus1, procenat1, odnosCene);
s->Izdvoji(sok, 2);
//s->print();
cout << *s << endl;

// 1 poena
Voda voda1(1);
s->Izdvoji(voda1, 2);
//s->print();
cout << *s << endl;

// 2 poena
Voda voda2 (2);
s->Presipaj(voda2, voda1);
//s->print();
cout << *s << endl;

// 2 poena
int nabavka = 7;
if (s->DovoljnaKolicina(nabavka))
    cout << "Ima dovoljno vode i sokova za " << nabavka << "1" << endl;
else
    cout << "Nema dovoljno vode i sokova" << endl;

// 2 poena
Pice* piceMin = nullptr, * piceMax = nullptr;
s->VratiNaj(piceMin, piceMax);
piceMin->Prikazi();
piceMax->Prikazi();

// 1 poena
delete s;
}

```

```

void zadatak2()
{
    int maxPodataka = 8, brojPodataka = 7;

    // 3 boda
    // postavljanje i prihvatanje izuzetaka
    {
        float voda[] = { 1, 0.2, 0.5, 1, 0.5, 0.2, 0.2 };
        Fabrika<float> fv(maxPodataka);
        // 1 bod
        for (int i = 0; i < brojPodataka; i++)
            fv.Dodaj(voda[i]);
        cout << fv << endl; // fv.Ispisi(cout);
        // 1 bod
        fv.Izbaci(voda[1]);
        cout << fv << endl; //fv.Ispisi(cout);
    }
}

```

```

// 1 bod
float ukupno = fv.Ukupno();
cout << ukupno << endl; //
// 1 bod
int vrednost = 1; float ukupnaVrednost = 0;
int broj = fv.UkupniBrojVrednost(vrednost, &ukupnaVrednost);
cout << broj << " " << ukupnaVrednost << endl; // 2 2
// 1 bod
int najmanjeAmbalaza = fv.NajmanjeAmbalaza(2.8);
cout << najmanjeAmbalaza << endl; // 4

// 1 bod
fv.Sacuvaj("NovaFabrikaVode.txt");
// 1 boda
Fabrika<float> novaUcitana(maxPodataka);
novaUcitana.Ucitaj("NovaFabrikaVode.txt");
cout << novaUcitana << endl; // novaUcitana.Ispisi(cout);
}

{
    char ukus[][7] = { "jagoda", "jabuka", "jagoda", "jagoda", "jagoda", "jabuka",
"jagoda" };
    float zap[] = { 1, 0.2, 0.5, 1, 0.5, 0.2, 0.2 };
    int procenat[] = { 50, 70, 50, 50, 50, 70, 50 };
    Fabrika<Sokovi> fs(maxPodataka);
    // 1 bod
    for (int i = 0; i < brojPodataka; i++) {
        fs.Dodaj(Sokovi(ukus[i], zap[i], procenat[i]));
    }
    // 1 bod
    cout << fs << endl; //fs.Ispisi(cout);
    // 1 bod
    fs.Izbaci(Sokovi(ukus[1], zap[1], procenat[1]));
    cout << fs << endl; //fs.Ispisi(cout);

    // 2 boda
    float ukupno = fs.Ukupno();
    cout << ukupno << endl;
    // 2 boda
    Sokovi vr(ukus[0], zap[0], procenat[0]);
    float ukupnaVrednost = 0;
    int broj = fs.UkupniBrojVrednost(vr, &ukupnaVrednost);
    cout << broj << " " << ukupnaVrednost << endl; // 2 2
    // 2 boda
    int najmanjeAmbalaza = fs.NajmanjeAmbalaza(2.5);
    cout << najmanjeAmbalaza << endl; // 3

    // 0 bodova
    fs.Sacuvaj("NovaFabrikaSokova.txt");
    // 1 bod
    Fabrika<Sokovi> novaUcS(maxPodataka);
    novaUcS.Ucitaj("NovaFabrikaSokova.txt");
    cout << novaUcS << endl; // novaUcS.Ispisi(cout);
}
}

void main()
{
    zadatak1();
    zadatak2();
}

```