

Први колоквијум из Објектно оријентисаног програмирања I

1) (укупно 70 поена) Написати на језику C++ следеће класе (класе опремити оним конструкторима и деструктором који су потребни за безбедно и ефикасно коришћење класа у општем случају):

- (20 поена) **Артикал** има назив, реалну цену и целобројни износ попуста (подразумевано 0). Подаци артикла се задају при стварању (није потребна провера). Може да се дохвати назив, цена са попустом и износ попуста. На главном излазу се испишује у облику `назив (цена_са_попустом)`.
- (20 поена) **Ставка** рачуна се креира за задати артикал и задату целобројну количину. Ставки рачуна се додељује редни број на рачуну приликом додавања ставке на рачун. Може да се израчуна износ ставке рачуна као производ количине и цене артикла. Може да се дохвати артикал ставке. На главном излазу се испишује у облику `S редни_број_на_рачуну (артикал) : количина | износ`.
- (30 поена) **Рачун** садржи произвољан број ставки и целобројни износ додатног попуста. Ствара се празан, након чега се ставке додају појединачно. Може да се постави износ додатног попуста. Може да се израчуна износ рачуна као сума износа свих појединачних ставки на рачуну. Уколико постоји додатни попуст, он се обрачунава само за артикле који већ нису на попусту. На главном излазу се испишује у облику `Racun : износ_рачуна`, а затим се у сваком реду испишује по једна ставка у облику `ставка [износ_са_додатним_попустом]`.

(0 поена) Приложена је главна функција која ствара један рачун и у њега дода неколико ставки са по једним артиклом. Након тога створи нови рачун као копију претходног, одобри се додатан попуст на нови рачун, а затим испише оригинални и нови рачун на главном излазу. Називе класа и метода прилагодити главној функцији, тако да се програм може превести и извршити.

НАПОМЕНЕ: а) Израда задатка траје 80 минута.

б) Рад се предаје на предвиђеном мрежном диску.

в) На располагању је документација на Web-у на адресама: <http://www.cplusplus.com/> и <http://en.cppreference.com>

г) Није дозвољено имати поред себе друге материјале, нити уз себе имати електронске уређаје, без обзира да ли су укључени или искључени.

д) Резултати колоквијума биће објављени на Web-у на адреси: <http://rti.etf.rs/rti/ir2001/index.html/>

```

#include <iostream>
#include <string>

class Artikal {
    std::string naziv;
    double cena;
    int popust;
public:
    Artikal(std::string n, double c, int p = 0) :
        naziv(n), cena(c), popust(p) {}

    std::string dohvNaziv() const {return naziv;}
    double dohvCena() const {
        return (100. - popust) / 100 * cena;
    }
    double dohvPopust() const { return popust; }
    void pisi() const {
        std::cout << naziv << "(" << dohvCena()
            << ")";
    }
};

class Stavka {
    Artikal artikal;
    int kolicina, rdBr;
public:
    Stavka(Artikal a, int k) : artikal(a),
        kolicina(k) {}

    void dodeliRdBr(int br) { rdBr = br; }
    double iznos() const {
        return kolicina * artikal.dohvCena();
    }
    Artikal dohvArtikal() const {return artikal;}
    void pisi() const {
        std::cout << "S" << rdBr << "(";
        artikal.pisi();
        std::cout << ")" << kolicina << "| "
            << iznos();
    }
};

class Racun {
    struct Elem {
        Stavka s;
        Elem *sled;
        Elem(Stavka &st) : s(st), sled(nullptr) {}
    };
    Elem *prvi = nullptr, *posl = nullptr;
    int popust = 0;
    int rdBrStavke = 0;

    double iznosPoStavci(Stavka &s) const {
        if (popust)
            return s.dohvArtikal().dohvPopust() == 0
                ? (s.iznos() * (100. - popust) / 100) :
                s.iznos();
        else return s.iznos();
    }
public:
    Racun() = default;
    Racun(const Racun &r);
    Racun(Racun &&r);
    ~Racun();
    void postaviPopust(int pop) { popust = pop; }
    Racun& dodaj(Stavka s);
    double iznos() const;
    void pisi() const;
};

Racun::Racun(const Racun &r) {
    Elem *tek = r.prvi;
    Elem *novi;
    while (tek) {
        novi = new Elem(tek->s);
        if (!prvi) prvi = novi;
        else posl->sled = novi;
        posl = novi;
        tek = tek->sled;
    }
}

Racun::Racun(Racun &&r) {
    prvi = r.prvi;
    posl = r.posl;
    popust = r.popust;
    rdBrStavke = r.rdBrStavke;
    r.prvi = r.posl = nullptr;
}

Racun::~Racun() {
    while (prvi) {
        Elem *tek = prvi;
        prvi = prvi->sled;
        delete tek;
    }
    posl = nullptr;
}

Racun& Racun::dodaj(Stavka s) {
    s.dodeliRdBr(++rdBrStavke);
    Elem *novi = new Elem(s);
    if (!prvi) prvi = novi;
    else posl->sled = novi;
    posl = novi;
    return *this;
}

double Racun::iznos() const {
    Elem *tek = prvi;
    double vrednost = 0;
    while (tek) {
        vrednost += iznosPoStavci(tek->s);
        tek = tek->sled;
    }
    return vrednost;
}

void Racun::pisi() const {
    std::cout << "Racun: " << iznos() <<
        std::endl;
    Elem *tek = prvi;
    while (tek) {
        tek->s.pisi();
        std::cout << " [" << iznosPoStavci(tek->s)
            << "]" << std::endl;
        tek = tek->sled;
    }
}

int main() {
    Artikal a1("Mleko", 100), a2("Hleb", 50),
        a3("Sir", 200, 20);
    Stavka s1(a1, 2), s2(a2, 3), s3(a3, 1);
    Racun r;
    r.dodaj(s1).dodaj(s2).dodaj(s3);
    Racun novi = r;
    novi.postaviPopust(20);
    r.pisi();
    novi.pisi();
}

Racun: 510
S1(Mleko(100)):2|200 [200]
S2(Hleb(50)):3|150 [150]
S3(Sir(160)):1|160 [160]
-----
Racun: 440
S1(Mleko(100)):2|200 [160]
S2(Hleb(50)):3|150 [120]
S3(Sir(160)):1|160 [160]

```