

Први колоквијум из
Објектно оријентисаног програмирања I

1) (30 поена) Одговорити концизно (по једна или две реченице) и прецизно на следећа питања:

а) Ако важи: `int a, b, c;` који изрази су *лвредности* (навести само редни број израза): (1) `a=b`
(2) `(a?b:c)` (3) `a--` (4) `++a` (5) `a+b?`

б) Шта исписује следећи програм написан на језику C++ ?

<pre>#include <iostream> using namespace std; class X { public: void m() & { cout << 1; } void m() && { cout << 2; } void m() const & { cout << 3; } void m() const && { cout << 4; } };</pre>	<pre>const X f1(X x){return x;} X f2(X x){return x;} int main() { const X x1; X x2; x1.m(); x2.m(); f1(x1).m(); f2(x1).m(); return 0; }</pre>
--	---

в) Шта је потребно обезбедити у премештајућем конструктору класе која реализује садржање по адреси (показивачу)?

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

- (20 поена) **Датум** има целобројне дан, месец и годину. Компоненте датума се задају при стварању (није потребна провера) и могу да се дохвате појединачно. Може да се провери да ли је датум већи од задатог датума. Датум може да се испише на главном излазу у облику *дан .месец .година*.
- (20 поена) **Лек** има једнозначни аутоматски генерисани идентификатор, комерцијално име, генерички назив, рок трајања (датум до којег се може користити) и цену. Сви подаци се задају приликом стварања и могу да се дохвате, а само цена може да се промени. Приликом копирања лека идентификатор се не копира. На главном излазу се исписује у облику *идентификатор-комерцијано име (генерички назив) :рок-цена*.
- (20 поена) **Апотека** има име и садржи произвољан број лекова. Ствара се празна, после чега се лекови додају појединачно. Не може да се направи копија апотеке, а при уништавању се не уништавају садржани лекови. Може да се дохвати лек по задатом идентификатору. На главном излазу се исписује тако што се у првом реду испише име апотеке, а потом се у засебним редовима испишу садржани лекови.

(10 поена) Написати на језику C++ **програм** који створи једну апотеку и у њу дода неколико лекова, испише је на главном излазу, а затим дохвати лек са идентификатором 3 и провери да ли је у року трајања на дан 04.11.2016. Користити фиксне параметре – није потребно ништа учитати с главног улаза.

НАПОМЕНЕ: а) Колоквијум траје **100** минута.

б) Рад се предаје искључиво у вежбанци за испите (-5 поена за неадекватну вежбанку). Није дозвољено имати поред себе друге листове папира, нити уз себе имати мобилни телефон, без обзира да ли је укључен или искључен.

в) Водити рачуна о уредности. Нечитки делови текста ће бити третирани као непостојећи. Решења задатака навести по горњем редоследу (-1 поен за лош редослед). Препоручује се рад обичном графитном оловком.

г) Решење задатка не треба раздвајати у датотеке. Довољно је за сваку класу навести дефиницију класе и одмах иза ње евентуалне дефиниције метода које нису дефинисане у самој класи.

д) Резултати колоквијума биће објављени на Web-у на адреси:
<http://rti.etf.bg.ac.rs/rti/ir2ool/index.html/>

```

#include <iostream>
#include <string>
using namespace std;
class Datum {
public:
    Datum(int dd, int mm, int gg)
    { d = dd; m = mm; g = gg; }
    int dan() const { return d; }
    int mesec() const { return m; }
    int godina() const { return g; }
    friend bool veci(const Datum&,
                     const Datum&);

    void pisi() const
    { cout << d << "." << m << "." << g
      << "."; }

private:
    int d, m, g;
};
bool veci(const Datum& d1,
          const Datum& d2) {
    if (d1.g > d2.g) return true;
    if (d1.m > d2.m) return true;
    return d1.d > d2.d;
}

class Lek {
public:
    Lek(string iime, string ggNaziv,
         Datum rrok, double ccena):
        rok(rrok) {
            ime = iime; gNaziv = ggNaziv;
            cena = ccena;
        }
    Lek(const Lek& l) : rok(l.rok) {
        ime = l.ime; gNaziv = l.gNaziv;
        cena = l.cena;
    }
    int dohvId() const { return id; }
    string dohvIme() const {return ime;}
    string dohvGNaziv() const {
        return gNaziv; }
    Datum dohvRok() const {return rok;}
    void promeniCenu(double nCena) {
        cena = nCena; }

    void pisi() const;

private:
    static int ID; int id = ++ID;
    string ime, gNaziv;
    Datum rok; double cena;
};
int Lek::ID = 0;
void Lek::pisi() const
{ cout << id << "-" << ime << "(" <<
  gNaziv << "):";
  rok.pisi(); cout << "-" << cena; }

class Apoteka {
public:
    Apoteka(string iime)
    { ime=iime; prvi=posl=nullptr; }
    Apoteka(const Apoteka&) = delete;
    Apoteka& dodaj(Lek* l) {
        posl = (!prvi ? prvi : posl->sled)
              = new Elem(l);

        return *this;
    }

```

```

    Lek* dohvLek(int id);
    const Lek* dohvLek(int id) const;
    void pisi() const;
    ~Apoteka();
private:
    struct Elem {
        Lek *lek; Elem *sled;
        Elem(Lek *l, Elem *s = nullptr) {
            lek = l; sled = s;
        }
    };
    Elem *prvi, *posl; string ime;
};
Lek* Apoteka::dohvLek(int id) {
    Elem *tek = prvi;
    while (tek != nullptr) {
        if (tek->lek->dohvId() == id)
            return tek->lek;
        tek = tek->sled;
    }
    return nullptr;
}
const Lek* Apoteka::dohvLek(int id)
    const {
    return const_cast<const Lek*>
        (const_cast<Apoteka&>(*this).
         dohvLek(id));
}
void Apoteka::pisi() const {
    cout << ime << ":" << endl;
    Elem *tek = prvi;
    while (tek != nullptr) {
        tek->lek->pisi(); cout << endl;
        tek = tek->sled;
    }
}
Apoteka::~~Apoteka() {
    Elem *tek = prvi;
    while (tek != nullptr) {
        Elem *stari = tek;
        tek = tek->sled; delete stari;
    }
    prvi=nullptr; posl=nullptr;
    ime="";
}

int main() {
    Apoteka a("Zdravlje");
    Lek l1("Diklofen", "Diklofenak",
           Datum(18, 5, 2017), 450),
        l2("Atolip", "Atovarstatin",
           Datum(2, 8, 2018), 250),
        l3("Alfacet", "Cefaklor",
           Datum(5, 11, 2016), 300),
        l4("Dovicin", "Doksiciklin",
           Datum(5, 11, 2012), 300);
    a.dodaj(&l1).dodaj(&l2).dodaj(&l3)
      .dodaj(&l4).pisi();
    if (veci((*a.dohvLek(3)).dohvRok(),
             Datum(4, 11, 2016)))
        cout << "Jeste!";
    else cout << "Nije!"; return 0;
}

Zdravlje:
1-Diklofen(Diklofenak):18.5.2017.-450
2-Atolip(Atovarstatin):2.8.2018.-250
3-Alfacet(Cefaklor):5.11.2016.-300
4-Dovicin(Doksiciklin):5.11.2012.-300

```