

Први колоквијум из Објектно оријентисаног програмирања I

- 1) (30 поена) Одговорити концизно (по једна или две реченице) и прецизно на следећа питања:
- а) Навести разлику између појма декларације и дефиниције функције у језику.
 - б) Којим операторима и на који начин се управља животним веком динамичког појединачног објекта? Шта су операнди, а шта резултати операција?
 - в) Да ли могу постојати две истоимене методе са идентичним листама параметара (формалних аргумената), које се разликују само по модификатору `const` иза листе аргумената и зашто?
- 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) **Тачка** има две задате реалне координате. Може да се одреди растојање од задате тачке (корен збира квадрата разлика у координатама), подразумевано од координатног почетка. Исписује се на главном излазу у облику (x, y) .
 - (20 поена) **Троугао** садржи адресе три задате тачке (темена) и задато име (`string`). Темена могу да се дохвате по редном броју (почевши од 0). Дохваћено теме не може да се мења. Троуглу може да се израчуна средње (аритметичка средина) растојање темена од координатног почетка. При копирању троугла не копирају се темена већ само њихове адресе, а на име оригинала се додаје "`_копија`". Исписује се на главном излазу у облику `име[теме0,теме1,теме2: средње_растојање]`.
 - (20 поена) **Мрежа** се састоји од произвољног броја троуглова и има задато име (`string`). Ствара се празна, троуглови се додају појединачно. Може да се направи нова мрежа од троуглова у задатом опсегу средњег растојања темена од координатног почетка. Име нове мреже је име оригинала са наставком "`_део`". Мрежа може да се испише на главном излазу тако што се у једном реду испише име мреже, а затим у посебним редовима троуглови мреже.
- (10 поена) Написати на језику C++ **програм** који направи неколико тачака и троуглова које дода у празну мрежу, те испише мрежу на главном излазу. Затим, за задати опсег средњег растојања темена од координатног почетка, направи нову мрежу и испише је на главном излазу. 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 третирано 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/ir2001/index.html/>

```

#include <iostream>
#include <string>
#include <cmath>
#include <cstdlib>
using namespace std;
class Tacka {
    double x, y;
public:
    Tacka(double xx, double yy)
        { x = xx; y = yy; }
    double rast(const Tacka& t=
                Tacka(0, 0))const {
        return sqrt((x-t.x)*(x-t.x)
                    +(y-t.y)*(y-t.y));
    }
    void pisi() const
        { cout <<'('<<x<<', '<<y<<')'; }
};

class Trougao{
    Tacka* tem[3];    string ime;
public:
    Trougao(Tacka* t[3], string i);
    Trougao(const Trougao& t);
    // Trougao(Trougao&& t) = default;
    const Tacka& dohTeme(int ind)const {
        if((ind<0)|| (ind>2)) exit(1);
        return *tem[ind];
    }
    double srRast() const;
    void pisi() const;
};

Trougao::Trougao(
    Tacka* t[3], string i): ime(i)
{ for(int i=0;i<3;i++)tem[i]=t[i]; }
Trougao::Trougao(const Trougao& t) {
    for(int i=0;i<3;i++)tem[i]=t.tem[i];
    ime = t.ime + "_kopija";
}
double Trougao::srRast() const {
    return (tem[0]->rast()+
            tem[1]->rast()+tem[2]->rast())/3;
}
void Trougao::pisi() const {
    cout << ime << "[";
    tem[0]->pisi(); cout << ", ";
    tem[1]->pisi(); cout << ", ";
    tem[2]->pisi();
    cout <<": "<<srRast()<<"]"<< endl;
}

class Mreza{
    struct Elem {
        Trougao t; Elem* sled;
        Elem(const Trougao& tt, Elem* ss=
              nullptr):t(tt),sled(ss){}
    };
    Elem *prvi, *posl; string ime;
public:
    Mreza(string i)
        { prvi=posl=nullptr; ime=i; }
    Mreza(const Mreza& m);
    Mreza(Mreza&& m) {
        prvi=m.prvi; posl=m.posl;
        ime=m.ime; m.prvi=m.posl=nullptr;
    }
};

```

```

~Mreza();
Mreza& dodaj(const Trougao& t);
Mreza napraviNovu(double min,
                  double max) const;
void pisi() const;
};

Mreza::Mreza(const Mreza& m){
    prvi = posl = nullptr;
    for(Elem* tek=m.prvi;
        tek; tek=tek->sled){
        posl=(!prvi ? prvi : posl->sled)
            = new Elem(tek->t);
    }
}

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

Mreza& Mreza::dodaj(const Trougao& t){
    posl=(!prvi?prvi:posl->sled)
        =new Elem(t);
    return *this;
}

Mreza Mreza::napraviNovu
    ( double min, double max ) const {
    if (min > max) exit(2);
    Mreza m = Mreza(ime+"_deo");
    Elem *tek = prvi;
    while(tek){
        if( min<=tek->t.srRast() &&
            max>=tek->t.srRast() )
            m.dodaj(tek->t);
        tek=tek->sled;
    }
    return m;
}

void Mreza::pisi() const {
    cout << ime << endl;
    for(Elem* tek=prvi; tek;
        tek=tek->sled)
        { tek->t.pisi(); cout << endl; }
}

int main() {
    Mreza m("Mreza");
    Tacka t1(1,1),t2(2,2),t3(3,3),
           t4(1,2),t5(3,4);
    Tacka *niz1[] = { &t1, &t2, &t3 };
    Tacka *niz2[] = { &t1, &t4, &t5 };
    Tacka *niz3[] = { &t3, &t4, &t5 };
    m.dodaj(Trougao(niz1, "t1"));
    m.dodaj(Trougao(niz2, "t2"));
    m.dodaj(Trougao(niz3, "t3"));
    m.pisi();
    Mreza novaM = m.napraviNovu(2, 3);
    novaM.pisi();
    return 0;
}

Izlaz:Mreza
t1_kopija[(1,1), (2,2), (3,3): 2.82843]
t2_kopija[(1,1), (1,2), (3,4): 2.88343]
t3_kopija[(3,3), (1,2), (3,4): 3.82624]
Mreza_deo
t1_kopija_kopija[(1,1), (2,2), (3,3): 2.82843]
t2_kopija_kopija[(1,1), (1,2), (3,4): 2.88343]

```