ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за СИ

Београд 27.10.2017.

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):

- *(15 поена)* ***Извођач*** има задато име и задат музички жанр. Жанр може бити *POP*, *REP* или *ROK* и може да се дохвати. На главном излазу се исписује у облику *име* **(***жанр***)**.
- *(25 поена)* ***Песма*** има назив, трајање изражено бројем минута и бројем секунди и максимални број извођача. Сви подаци се задају приликом стварања. Може да се дохвати број минута и број секунди трајања песме, да се провери која од две задате песме траје дуже, као и да се додају појединачни извођачи (додавање не успева у случају прекорачења капацитета). Песма се не може копирати ни на који начин. На главном излазу се исписује у облику **P(***назив* **–** *минути* **:** *секунди***)**, а затим се појединачни извођачи исписују у засебним редовима.
- *(30 поена)* ***Плејлиста*** има трајање изражено бројем минута и бројем секунди и садржи произвољан број песама. Ствара се празна, након чега се песме додају појединачно уз ажурирање трајања плејлисте. Плејлиста може да се уреди по растућим временима трајања њених песама. Приликом уништавања плејлисте не уништавају се њене песме. На главном излазу се исписује у облику **Lista - trajanje:** *минути* **:** *секунди*, а затим се у сваком реду исписује по једна песма.

(0 поена) Приложена је главна функција која ствара једну плејлисту и у њу дода неколико песама са неколико извођача. Након тога створи нову плејлисту као копију претходне, уреди нову плејлисту, а затим испише оригиналну и нову плејлисту 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/ir2oo1/index.html/`

```
#include <iostream>
#include <string>

using namespace std;

enum Zanr { POP, REP, ROK };

class Izvodjac {
   string naziv;
   Zanr zanr;
   static string str_zanr[];
public:
   Izvodjac(string naz, Zanr z) :
      naziv(naz), zanr(z) {}
   Zanr dohvatiZanr() const { return zanr; }
   void pisi() const;
};

void Izvodjac::pisi() const {
   cout << naziv << "("
        << str_zanr[zanr] << ")" << endl;
}
string Izvodjac::str_zanr[]
                  = { "pop", "rep", "rok" };
```

```
class Pesma {
   int min, sek;
   string naziv;
   Izvodjac **izv;
   int br = 0, kap;
public:
   Pesma(int m, int s, string naz, int k) :
      min(m), sek(s), naziv(naz) {
      izv = new Izvodjac* [kap = k];
   }
   Pesma(const Pesma&) = delete;
   ~Pesma() { delete [] izv; }

   void dodaj(Izvodjac *i) {
      if (br < kap) izv[br++] = i;
   }
   int dohvSek() const { return sek; }
   int dohvMin() const { return min; }
   void pisi() const;
   friend bool duze(const Pesma &p1,
                    const Pesma &p2) {
      if (p1.min > p2.min ||
          p1.min == p2.min && p1.sek > p2.sek)
           return true;
      else return false;
   }
};

void Pesma::pisi() const {
   cout << "P(" << naziv << " - "
        << min << ":" << sek << ")" << endl;
   cout << "Izvodjaci: ";
   for (int i = 0; i < br; i++) izv[i]->pisi();
}
```

```
class Plejlista {
   struct Elem {
      Pesma *pes;
      Elem *sled;
      Elem(Pesma *p, Elem *s = nullptr) :
           pes(p), sled(s) {}
   };

   Elem *prvi = nullptr;
   int min = 0, sek = 0;
public:
   Plejlista() = default;
   Plejlista(const Plejlista &p);
   Plejlista(Plejlista &&p);
   ~Plejlista();
   void dodaj(Pesma *p);
   void uredi();
   void pisi() const;
};
```

```
Plejlista::Plejlista(const Plejlista &p) {
   min = p.min;
   sek = p.sek;
   for (Elem *tek = p.prvi; tek; tek=tek->sled)
      prvi = new Elem(tek->pes, prvi);
}
Plejlista::Plejlista(Plejlista &&p) {
   min = p.min;
   sek = p.sek;
   prvi = p.prvi;
   p.prvi = nullptr;
}
Plejlista::~Plejlista() {
   Elem *tek = prvi, *stari;
   while (tek) {
      stari = tek;
      tek = tek->sled;
      delete stari;
   }
}
void Plejlista::dodaj(Pesma *p) {
   prvi = new Elem(p, prvi);
   min = min + p->dohvMin() +
   (sek + p->dohvSek()) / 60;
   sek = (sek + p->dohvSek()) % 60;
}
void Plejlista::uredi() {
   for (Elem *t1=prvi; t1->sled; t1=t1->sled)
      for (Elem *t2 = t1->sled; t2; t2=t2->sled)
         if (duze(*t1->pes, *t2->pes)) {
            Pesma *p = t1->pes;
            t1->pes = t2->pes;
            t2->pes = p;
         }
}
void Plejlista::pisi() const {
   cout << "Lista - trajanje: " << min << ":" <<
   sek << endl;
   for (Elem *tek = prvi; tek; tek = tek->sled)
      tek->pes->pisi();
}
```

```
int main() {
   Izvodjac iz1("Micko", Zanr::POP), iz2("Uki",
   Zanr::ROK), iz3("Jocke", Zanr::REP);
   Pesma p1(2, 55, "Pesma", 2);
   Pesma p2(3, 23, "Pesma2", 1);
   Pesma p3(2, 49, "Pesma3", 1);
   Plejlista pl;

   p1.dodaj(&iz2); p1.dodaj(&iz3);
   p2.dodaj(&iz1); p3.dodaj(&iz2);
   pl.dodaj(&p1); pl.dodaj(&p2); pl.dodaj(&p3);

   Plejlista p = pl;
   p.uredi();

   pl.pisi();
   cout << "-------------------------" << endl;
   p.pisi();
}
```

```
Lista - trajanje: 9:7
P(Pesma3 - 2:49)
Izvodjaci: Uki(rok)
P(Pesma2 - 3:23)
Izvodjaci: Micko(pop)
P(Pesma - 2:55)
Izvodjaci: Uki(rok)
Jocke(rep)
-------------------------
Lista - trajanje: 9:7
P(Pesma3 - 2:49)
Izvodjaci: Uki(rok)
P(Pesma - 2:55)
Izvodjaci: Uki(rok)
Jocke(rep)
P(Pesma2 - 3:23)
Izvodjaci: Micko(pop)
```