

UNIVERZITET U TUZLI  
FAKULTET ELEKTROTEHNIKE



# Objektno-orijentirano programiranje

Zadaća 3

Tuzla, decembar/prosinac 2019.

# Sadržaj

<b>Sadržaj</b>	<b>2</b>
<b>Zadatak 1</b>	<b>3</b>
<b>Zadatak 2</b>	<b>3</b>
<b>Zadatak 3</b>	<b>3</b>

## Zadatak 1

Napisati generičku funkciju `jednako` koja poredi sekvence  $[od1, do1)$  i  $[od2, do2)$ , pri čemu se pretpostavlja da sekvence imaju isti broj elemenata. Funkcija vraća vrijednost `true` ukoliko su svi elementi obje sekvence isti, u suprotnom vraća netačnu vrijednost. Funkcija uzima tri parametra - prva dva označavaju početak i kraj prve sekvence, a treći parametar označava početak druge sekvence.

Napomena: iteratori koji operiraju na sekvencama mogu biti različitog tipa.

Koristeći funkciju `jednako`, napisati program koji učitava riječi koje korisnik unosi sa tastature i ispisuje svaku riječ koju identificira kao palindrom. Palindrom je riječ koja ima isto značenje bez obzira da li se čita sa lijeva na desno ili obrnuto. Primjer pozivanja programa se nalazi na [linku](#).

## Zadatak 2

Implementirati generičku funkciju `sort(p, k, f)` koja kao argumente uzima početak i kraj sekvence koju treba sortirati  $[p, k)$  te binarnu predikat funkciju `f(a, b)` koja definiše kriterij za sortiranje. Binarna predikat funkcija vraća tačnu vrijednost ukoliko element `a` treba u sekvenci biti prije elementa `b`, u suprotnom vraća netačno. Ukoliko se funkcija `sort` pozove bez trećeg argumenta (pozove se `sort(p, k)`), elemente sortirati od manjeg ka većem. Funkciju `sort` definisati u namespace-u `my`, u zaglavlju `my.hpp`.

Pseudo-kod algoritma za sortiranje kojeg implementira funkcija `sort`, skupa sa primjerima koji uključuju animaciju može se naći na adresi: <https://visualgo.net/en/sorting>.

Funkciju testirati pomoću file-a `test.cpp` datog u prilogu zadaće. Obratiti pažnju na činjenicu da file-ovi `test.cpp`, `my.hpp` i `doctest.h` (nalazi se u prilogu zadaće) moraju biti u istom direktoriju. Na [linku](#) se nalazi poziv izvršnog file-a sa ispravno napisanom funkcijom `sort`.

## Zadatak 3

Implementirati klase `Radnik` i `BazaRadnika` koja implementira bazu radnika i omogućava:

- Unos pojedinačnog radnika u bazu. Baza ne dozvoljava unos dva radnika istog imena. Ukoliko se pokuša unijeti isti radnik dva puta, metod generira iznimku tipa `domain_error`.
- Brisanje radnika određenog imena iz baze. Metod za brisanje kao rezultat vraća uspješnost operacije.
- Unos i ispis kompletne baze pomoću objekata tipa `istream` i `ostream`.
- Računanje prosječne plate i starosti radnika.

Klasu testirati sljedećim glavnim programom:

```
#include "Radnik.hpp"
#include "Baza.hpp"
#include <iostream>

int main() {
    BazaRadnika b1;
    b1.ucitaj(std::cin);
    if(b1.izbrisi_radnika("Alen")) {
        std::cout << "Alen izbrisan" << std::endl;
    } else {
        std::cout << "Brisanje nije moguće!" << std::endl;
    }
    std::cout << "U bazi su: " << std::endl;
    b1.ispisi(std::cout);
    std::cout<<"Prosjek plata je: "<<b1.prosjek_plata()<<std::endl;
    std::cout<<"Prosjek godina je: "<<b1.prosjek_godina()<<std::endl;
    return 0;
}
```

Na [linku](#) je prikazan poziv izvršnog file-a sa navedenom main funkcijom.