Zadaca 1

# Zadatak 1.

#include <iostream>

#include <string>

using namespace std;

struct TPerson

{

string name;

string surname;

int age;

string phoneNumber;

string address;

char gender;

long long int ID;

};

void unos(TPerson &person)

{

cout << "Name = ? ";

getline(cin, person.name);

cout << "Surname = ? ";

getline(cin, person.surname);

cout << "Age = ? ";

cin >> person.age;

cin.ignore();

cout << "Phone Number = ? ";

getline(cin, person.phoneNumber);

cout << "Address = ? ";

getline(cin, person.address);

cout << "Gender = ? ";

cin >> person.gender;

cout << "ID = ? ";

cin >> person.ID;

}

void ispis(TPerson &person)

{

cout << endl;

cout << "Name is " << person.name << " " << person.surname << endl;

cout << "Phone Number is " << person.phoneNumber << endl;

cout << "Address is " << person.address << endl;

cout << "Age is " << person.age << endl;

cout << "Gender is " << person.gender << endl;

cout << "ID is " << person.ID << endl;

}

int main()

{

TPerson person;

unos(person);

ispis(person);

return 0;

}

# Zadatak 2.

#include <iostream>

#include <string>

using namespace std;

class Bankovni\_racun

{

private:

string broj\_racuna;

double stanje\_na\_racunu;

double kamatna\_stopa;

public:

Bankovni\_racun (string b, double s, double k)

{

broj\_racuna = b;

stanje\_na\_racunu = s;

kamatna\_stopa = k;

}

double Izracunaj\_kamatu ();

double Daj\_Stanje() const;

string Daj\_Broj\_Racuna() const;

void Uplata (double);

bool Isplata (double);

};

double Bankovni\_racun::Izracunaj\_kamatu()

{

double kamata;

kamata = kamatna\_stopa \* stanje\_na\_racunu;

return kamata;

}

double Bankovni\_racun::Daj\_Stanje () const

{

return stanje\_na\_racunu;

}

string Bankovni\_racun::Daj\_Broj\_Racuna () const

{

return broj\_racuna;

}

void Bankovni\_racun::Uplata (double x)

{

stanje\_na\_racunu += x;

}

bool Bankovni\_racun::Isplata (double x)

{

if (stanje\_na\_racunu >= x)

{

stanje\_na\_racunu -= x;

cout << "moguca" << endl;

}

else

{

cout << "nemoguca" << endl;

cout << "Razlog: nedovoljan iznos na racunu" << endl;

}

}

int main ()

{

double u, i;

Bankovni\_racun korisnik("123-000-000", 7500, 0.2);

cout << "Kamata: " << korisnik.Izracunaj\_kamatu() << endl;

cout << "Trenutno stanje racuna: " << korisnik.Daj\_Stanje() << endl;

cout << "Broj racuna: " << korisnik.Daj\_Broj\_Racuna() << endl;

cout << "Unesite sumu novca koju zelite uplatiti na racun: " << endl;

cin >> u;

korisnik.Uplata(u);

cout << "Trenutno stanje racuna: " << korisnik.Daj\_Stanje() << endl;

cout << "Unesite sumu novca koju zelite podici sa racuna: " << endl;

cin >> i;

cout << "Isplata: ";

cout << korisnik.Isplata(i);

cout << endl << "Trenutno stanje racuna: " << korisnik.Daj\_Stanje() << endl << endl;

return 0;

}

# Zadatak 3.

#include <iostream>

#include <string>

using namespace std;

class Bankovni\_racun

{

private:

string broj\_racuna;

double stanje\_na\_racunu;

double kamatna\_stopa;

public:

Bankovni\_racun (string b, double s, double k)

{

broj\_racuna = b;

stanje\_na\_racunu = s;

kamatna\_stopa = k;

}

Bankovni\_racun (string b, double s)

{

broj\_racuna = b;

stanje\_na\_racunu = s;

kamatna\_stopa = 0.02;

}

Bankovni\_racun (string b)

{

broj\_racuna = b;

stanje\_na\_racunu = 0;

kamatna\_stopa = 0.02;

}

double Izracunaj\_kamatu ();

double Daj\_Stanje() const;

string Daj\_Broj\_Racuna() const;

void Uplata (double);

bool Isplata (double);

};

double Bankovni\_racun::Izracunaj\_kamatu()

{

double kamata;

kamata = kamatna\_stopa \* stanje\_na\_racunu;

return kamata;

}

double Bankovni\_racun::Daj\_Stanje () const

{

return stanje\_na\_racunu;

}

string Bankovni\_racun::Daj\_Broj\_Racuna () const

{

return broj\_racuna;

}

void Bankovni\_racun::Uplata (double x)

{

stanje\_na\_racunu += x;

}

bool Bankovni\_racun::Isplata (double x)

{

if (stanje\_na\_racunu >= x)

{

stanje\_na\_racunu -= x;

cout << "moguca" << endl;

}

else

{

cout << "nemoguca" << endl;

cout << "Razlog: nedovoljan iznos na racunu" << endl;

}

}

int main ()

{

double u, i;

Bankovni\_racun korisnik("123-000-000", 7500, 0.2);

cout << "Kamata: " << korisnik.Izracunaj\_kamatu() << endl;

cout << "Trenutno stanje racuna: " << korisnik.Daj\_Stanje() << endl;

cout << "Broj racuna: " << korisnik.Daj\_Broj\_Racuna() << endl;

cout << "Unesite sumu novca koju zelite uplatiti na racun: " << endl;

cin >> u;

korisnik.Uplata(u);

cout << "Trenutno stanje racuna: " << korisnik.Daj\_Stanje() << endl;

cout << "Unesite sumu novca koju zelite podici sa racuna: " << endl;

cin >> i;

cout << "Isplata: ";

cout << korisnik.Isplata(i);

cout << endl << "Trenutno stanje racuna: " << korisnik.Daj\_Stanje() << endl << endl;

//----------------------------------------------------------------------------------------------------------------------------

Bankovni\_racun korisnik\_2("124-000-000", 1000);

cout << "Kamata: " << korisnik\_2.Izracunaj\_kamatu() << endl;

cout << "trenutno stanje racuna: " << korisnik\_2.Daj\_Stanje() << endl;

cout << "Broj racuna: " << korisnik\_2.Daj\_Broj\_Racuna() << endl<<endl;

//----------------------------------------------------------------------------------------------------------------------------

Bankovni\_racun korisnik\_3("125-000-000");

cout << "Kamata: " << korisnik\_3.Izracunaj\_kamatu() << endl;

cout << "trenutno stanje racuna: " << korisnik\_3.Daj\_Stanje() << endl;

cout << "Broj racuna: " << korisnik\_3.Daj\_Broj\_Racuna() << endl;

return 0;

}

# Zadatak 4.

Prvo ispisuje 5 5 jer je x=5, jer u konstruktoru sa jedinm parametrom vidimo da su a,b= x. A ispisuju se a i b.

A potom ispisuje 3 9, jer u konstruktoru s jednim parametrom vidimo da je a=x-odnosno prvi broj 3, a b=y-drugi broj 9. Funkcija print ispisuje a i b.

# Zadatak 5.

#include <iostream>

#include <cmath>

#include <vector>

using namespace std;

class Tacka

{

private:

double x, y;

public:

void setX(double);

void setY(double);

void Output() const;

double distance();

friend Tacka Najudaljenija\_tacka(vector <Tacka> vektor)

{

double dmax = 0;

int m = 0;

for (int i = 0; i<vektor.size(); i++)

{

if (vektor[i].distance() > dmax)

{

dmax = vektor[i].distance();

m = i;

}

}

return vektor[m];

}

};

void Tacka::setX(double a)

{

x = a;

}

void Tacka::setY(double a)

{

y = a;

}

void Tacka::Output() const

{

cout << "(" << x << ", " << y << ")" << endl;

}

double Tacka::distance()

{

return sqrt(x\*x + y\*y);

}

int main()

{

vector <Tacka> v1;

int velicina\_vektora;

Tacka MAX;

cout << "Unesite broj tacaka" << endl;

cin >> velicina\_vektora;

for(int i=0; i<velicina\_vektora; i++)

{

Tacka t;

double x1, y1;

cout << "Unesite kordinatu x " << i + 1 << ". tacke:" << endl;

cin >> x1;

t.setX(x1);

cout << "Unesite kordinatu y " << i + 1 << ". tacke:" << endl;

cin >> y1;

t.setY(y1);

v1.push\_back(t);

}

MAX = Najudaljenija\_tacka(v1);

MAX.Output();

return 0;