**SPREMINJANJE PODATKOV V DATOTEKI Z DIREKTNIM DOSTOPOM (random access)**

Brisanje podatkov iz datoteke oz. spreminjanje le-teh, lahko izvajamo tudi z direktnim dostopom.

**fstream** namesto ifstream in ofstream?

fstream združuje oba datotečna tokova v enem in je zato **nujno** pri postavljanju na datoteko **eksplicitno** zapisati, ali bomo tok uporabljali za zapis v datoteke ali za branje iz datoteke oz. zastavice in, out, app. Enako pomembna je **pravilna** uporaba metod seekg() in seekp().

Primer:

V datoteki ***osebe.bin*** želimo spremeniti podatke 3. osebe (3. zapisa) brez ustvarjanja nove datoteke, v katero bi prepisovali podatke, in brez uporabljanja remove() in rename().

#include <iostream>

#include <cstdlib>

#include <fstream>

#include <cstring>

using namespace std;

struct Oseba

{

char priimek[21];

int starost;

};

void izpis\_datoteke()

{

struct Oseba a;

fstream data("osebe.bin",ios::binary | ios::in);

if(data.is\_open())

{

cout<<"v datoteki so zapisi:\n ";

while(data.read((char\*)&a, sizeof(a)))

cout<<a.priimek<<", "<<a.starost<<endl;

data.close();

}

}

void spremeni()

{

fstream data("osebe.bin",ios::binary | ios::out | ios::in);

struct Oseba a;

if(data.is\_open())

{

data.seekg(2\*sizeof(a),ios::beg);

data.read((char\*)&a, sizeof(a));

cout<<"3. zapis je: "<<a.priimek<<", "<<a.starost<<endl;

cout<<"vpisi spremenjeno starost: \n";

cin>>a.starost;

data.seekp(2\*sizeof(a),ios::beg);

data.write((char\*)&a, sizeof(a));

data.close();

}

}

void vpis\_v\_datoteko()

{

fstream data("osebe.bin",ios::binary | ios::app);

struct Oseba a;

for(int i=0;i<5;i++)

{

fflush(stdin);

gets(a.priimek);

cin>>a.starost;

data.write((char\*)&a, sizeof(a));

}

data.close();

}

int main()

{

vpis\_v\_datoteko();

spremeni();

izpis\_datoteke();

return 0;

}