Strukture odvojite u zasebni header fajlove.

# Zadatak 1 - Datum

1. Deklarisati strukturu **Datum** sa atributima d, m i g.
2. Implementirati funkciju koja ispisuje podatke o datumu **date**

void info(Datum date)

1. Implementirati funkciju koja vraća objekat tipa Datum sa vrijednostima učitanih sa tastature

Datum ucitaj\_datum()

|  |
| --- |
| **Rješenje** |
| // stuktura Datum  struct Datum {  int d;  int m;  int g;  };  void info(Datum date) {  // dd/mm/gg  cout << date.d << "/" << date.m << "/" << date.g << endl;  }  Datum ucitaj\_datum()  {  Datum x;  cout << "unesite dan, mjesec i godinu: ";  cin >> x.d;  cin >> x.m;  cin >> x.g;  return x;  } |

# Zadatak 2 – Osoba

1. Deklarisati strukturu **Osoba** sa atributima id, ime.
2. Implementirati funkciju koja ispisuje podatke o osobi **o**

void info(Osoba o)

|  |
| --- |
| **Rješenje a i b** |
| // struktura Osoba  struct Osoba {  int id;  char ime[15];  };  void info(Osoba o) {  cout << "Ime osobe je: " << o.ime << " id= " << o.id << endl;  } |

1. Alocirajte 5 objekatat (kao globalne varijable) tipa osoba (o1, o2, o3, o4, o5) i dodjelite neke vrijednosti.
2. Alocirajte niz tipa osoba veličine 5.

const int max\_s = 5;

Osoba studenti[max\_s] = {o1, o2, o3, o4, o5}

|  |
| --- |
| **Rješenje c i d – sa varijablama o1, ... o5** |
| Osoba o1 = { 534, "Nina B." };  Osoba o2 = { 435, "Jasmin A."};  Osoba o3 = { 256, "Denis M." };  Osoba o4 = { 325, "Emina J." };  Osoba o5 = { 914, "Zanin V." };  const int max\_i = 5;  Osoba ispitivaci[max\_i] = { o1, o2, o3, o4, o5,}; |
| **Rješenje c i d – bez varijabli o1, ... o5** |
| const int max\_i = 5;  Osoba ispitivaci[max\_i] = {  { 534, "Nina B." },  { 435, "Jasmin A." },  { 256, "Denis M." },  { 325, "Emina J." },  { 914, "Zanin V." },  }; |

1. Implementirati funkciju koja vraća objekat iz niza **studenti** na osnovu rednog broja kojeg korisnik unese pomoću tastature.

|  |
| --- |
| Spisak osoba:  1. Osoba A B, ID=302  2. Osoba C B, ID=303  3. Osoba D B, ID=304  4. Osoba E B, ID=305  5. Osoba F B, ID=306  Unesite RB. |

Osoba odaberite\_osobu()

|  |
| --- |
| **Rješenje e** |
| Osoba odaberite\_osobu()  {  int a;  cout << "Spisak osoba: " << endl;  for (int i = 0; i < max\_s; i++)  {  cout << i << "Ime osobe: " << studenti[i].ime << ", ID: " <<  studenti[i].id << endl;  }  cout << "Unesite RB: ";  cin >> a;  Osoba x = studenti[a];  return x;  } |

1. U funkciji main implementirajte sljedeće

* Deklarište datume **d1**, **d2 i d3** i učitajte vrijednosti pomoću funkcije ucitaj\_datum
* Ispišite datume **d1, d2 i d3** pomoću funkcije info(datum)
* Odaberite studenta s1 i ispišite podatke pomoću funkcije info(osoba)

|  |
| --- |
| **Rješenje f** |
| Datum d1 = ucitaj\_datum();  Datum d2 = ucitaj\_datum();  Datum d3 = ucitaj\_datum();  info(d1);  info(d2);  info(d3);  Osoba s1 = odaberite\_osobu(studenti, max\_s);  info(s1); |

Prepravite prethodno rješenje

1. Prepravite funkciju odaberite\_osobu tako funkcija koristi različite nizove tipa osoba (npr. ispitivače i studente)

Osoba odaberite\_osobu() -> Osoba odaberite\_osobu(Osobe osobe[], int v)

|  |
| --- |
| **Rješenje g** |
| Osoba odaberite\_osobu(Osoba niz[], int z)  {  int a;  cout << "Spisak osoba: " << endl;  for (int i = 0; i < z; i++)  {  cout << i << "Ime osobe: " << niz[i].ime << ", ID: " <<  niz[i].id << endl;  }  cout << "Unesite RB: ";  cin >> a;  Osoba x = niz[a];  return x;  } |

# Zadatak 3 - Ispit

1. Implementirajte strukturu Ispit

|  |
| --- |
| struct Ispit  {  Osoba ispitivac;  Datum datum\_ispita;  char predmet[10];  };  Ispit ucitaj\_ispit();  void prikazi\_ispite(Ispit ispiti[], int max); |

|  |
| --- |
| **Rješenje** |
| struct Ispit {  Osoba ispitivac;  Datum datum\_ispita;  char predmet[10];  };  Ispit ucitaj\_ispit()  {  Ispit x;  cout << "Unesite predmet " << endl;  cin >> x.predmet;  cout << "Odaberite ispitivaca" << endl;  x.ispitivac = odaberite\_osobu(ispitivaci, max\_i);  cout << "Unesite datum ispita" << endl;  x.datum\_ispita = ucitaj\_datum();  return x;  }  void prikazi\_ispite(Ispit ispiti[], int max)  {  for (int i = 0; i < max; i++)  {  cout << i +1 << ". ";  info(ispiti[i].datum\_ispita);  info(ispiti[i].ispitivac);  cout << "Predmet -> " << ispiti[i].predmet << endl;  }  }  Ispit& odaberite\_ispit(Ispit ispiti[], int max)  {  prikazi\_ispite(ispiti, max);  cout << "Unesite RB ispita: ";  int x;  cin >> x;  return ispiti[x - 1];  } |

Program za testiranje

|  |
| --- |
| void main()  {  int \_brojac\_ispita = 0;  const int max\_ispita = 30;    Ispit ispiti[max\_ispita];  ispiti[\_brojac\_ispita++] = ucitaj\_ispit();  ispiti[\_brojac\_ispita++] = ucitaj\_ispit();  ispiti[\_brojac\_ispita++] = ucitaj\_ispit();  ispiti[\_brojac\_ispita++] = ucitaj\_ispit();  Ispit& i = odaberite\_ispit(ispiti, \_brojac\_ispita); //korisnik bira ispit  i.datum\_ispita.d++; //pomjeri datum ispita na naredni dan  prikazi\_ispite(ispiti, \_brojac\_ispita);  } |

# Zadatak 4 – Prijava za ispit

Dodajte strukturu **PrijavaZaIspit**.

U strukturu Ispit dodajte niz tipa **PrijavaZaIspit**

|  |
| --- |
| struct PrijavaZaIspit  {  Osoba student;  Datum datum\_prijave;  Datum datum\_odjave;  int ocjena;  };  const int max\_p = 100;  struct Ispit  {  Osoba ispitivac;  Datum datum\_ispita;  char predmet[10];  int brojac\_prijavljenih;  PrijavaZaIspit prijave[max\_p];  }; |

Prilagodite funkcijevoid ucitaj\_ispit() i void prikazi\_ispite(Ispit[], int) shodno novim modifikacijima u strukturi Ispit.

|  |
| --- |
| Rješenje |
| Ispit ucitaj\_ispit()  {  Ispit x;  cout << "Unesite predmet " << endl;  cin >> x.predmet;  cout << "Odaberite ispitivaca" << endl;  x.ispitivac = odaberite\_osobu(ispitivaci, max\_i);  cout << "Unesite datum ispita" << endl;  x.datum\_ispita = ucitaj\_datum();    x.brojac\_prijavljenih = 0;  return x;  }  void prikazi\_ispite(Ispit ispiti[], int max)  {  for (int i = 0; i < max; i++)  {  info(ispiti[i].datum\_ispita);  info(ispiti[i].ispitivac);  cout << "Predmet -> " << ispiti[i].predmet ;  cout << ", \tBroj prijavljenih: " << ispiti[i].brojac\_prijavljenih;  }  } |

# Zadatak 5 – Glavni program

Dovršite program

|  |
| --- |
| void main()  {  const int max\_ispiti = 10;  Ispit ispiti[max\_ispiti];  int brojac\_ispita = 0;  int x;  do  {  cout << "1. Ispiti - prikazi sve" << endl;  cout << "2. Ispiti - dodaj novi (zakazi novi ispit)" << endl;  cout << "3. Prijavljeni za ispit: prikazi sve (za ispit x)" << endl;  cout << "4. Prijavljeni za ispit: dodaj novu prijavu (za ispit x)" << endl;  cout << "5. Evidentiraj sve ocjene (za ispit x)" << endl;  cout << "0. EXIT" << endl;  cin >> x;  if (x == 1)  {  }  if (x == 2)  {    if (x == 3)  {  }  if (x == 4)  {  }  if (x == 5)  {  }  } while (x != 0);  } |

**Rješenje**

|  |
| --- |
| #include <iostream>  using namespace std;  // stuktura Datum  struct Datum {  int d;  int m;  int g;  };  void info(Datum date) {  // dd/mm/gg  cout << date.d << "/" << date.m << "/" << date.g << ". ";  }  Datum ucitaj\_datum()  {  Datum x;  cout << "unesite dan, mjesec i godinu: ";  cin >> x.d;  cin >> x.m;  cin >> x.g;  return x;  }  // struktura Osoba  struct Osoba {  int id;  char ime[15];  };  void info(Osoba o) {  cout << "Ime osobe je: " << o.ime << " id= " << o.id << ". ";  }  // Studenti  const int max\_s = 5;  Osoba studenti[max\_s] = {  { 1, "Marko" },  { 2, "Darko" },  { 3, "Zarko" },  { 4, "Adil" },  { 5, "Faruk" }  };  Osoba odaberite\_osobu(Osoba niz[], int z)  {  int a;  cout << "Spisak osoba: " << endl;  for (int i = 0; i < z; i++)  {  cout << i << "Ime osobe: " << niz[i].ime << ", ID: " << niz[i].id << endl;  }  cout << "Unesite RB: ";  cin >> a;  Osoba x = niz[a];  return x;  }  Osoba o1 = { 534, "Nina B." };  Osoba o2 = { 435, "Jasmin A."};  Osoba o3 = { 256, "Denis M." };  Osoba o4 = { 325, "Emina J." };  Osoba o5 = { 914, "Zanin V." };  const int max\_i = 5;  Osoba ispitivaci[max\_i] = { o1, o2, o3, o4, o5,};  struct PrijavaZaIspit  {  Osoba student;  Datum datum\_prijave;  Datum datum\_odjave;  int ocjena;  };  const int max\_p = 100;  struct Ispit {  Osoba ispitivac;  Datum datum\_ispita;  char predmet[10];  int brojac\_prijavljenih;  PrijavaZaIspit prijave[max\_p];  };  Ispit ucitaj\_ispit()  {  Ispit x;  cout << "Unesite predmet " << endl;  cin >> x.predmet;  cout << "Odaberite ispitivaca" << endl;  x.ispitivac = odaberite\_osobu(ispitivaci, max\_i);  cout << "Unesite datum ispita" << endl;  x.datum\_ispita = ucitaj\_datum();    x.brojac\_prijavljenih = 0;  return x;  }  void info(PrijavaZaIspit p)  {  cout << "Student: ";  info(p.student);  cout << " Datum prijave: ";  info(p.datum\_prijave);  cout << " Datum odjave: ";  info(p.datum\_odjave);  cout << endl;  }  void prikazi\_ispite(Ispit ispiti[], int max)  {  for (int i = 0; i < max; i++)  {  cout << i +1 << ". ";  info(ispiti[i].datum\_ispita);  info(ispiti[i].ispitivac);  cout << "Predmet -> " << ispiti[i].predmet ;  cout << ", \tBroj prijavljenih: " << ispiti[i].brojac\_prijavljenih << endl;  }  }  Ispit& odaberite\_ispit(Ispit ispiti[], int max)  {  prikazi\_ispite(ispiti, max);  cout << "Unesite RB ispita: ";  int x;  cin >> x;  return ispiti[x - 1];  }  void main()  {  const int max\_ispiti = 10;  Ispit ispiti[max\_ispiti];  int brojac\_ispita = 0;  int x;  do  {  cout << "1. Ispiti - prikazi sve" << endl;  cout << "2. Ispiti - dodaj novi (zakazi novi ispit)" << endl;  cout << "3. Prijavljeni za ispit: prikazi sve (za ispit x)" << endl;  cout << "4. Prijavljeni za ispit: dodaj novu prijavu (za ispit x)" << endl;  cout << "5. Evidentiraj sve ocjene (za ispit x)" << endl;  cout << "0. EXIT" << endl;  cin >> x;  if (x == 1)  {  prikazi\_ispite(ispiti, brojac\_ispita);  }  if (x == 2)  {  Ispit i = ucitaj\_ispit();  ispiti[brojac\_ispita] = i;  brojac\_ispita++;  }  if (x == 3)  {  Ispit i = odaberite\_ispit(ispiti, brojac\_ispita);  for (int j = 0; j<i.brojac\_prijavljenih; j++)  {  PrijavaZaIspit& p = i.prijave[j];  info(p);  }  }  if (x == 4)  {  Ispit& i = odaberite\_ispit(ispiti, brojac\_ispita);  PrijavaZaIspit novaP;  novaP.student = odaberite\_osobu(studenti, max\_s);  novaP.datum\_prijave = ucitaj\_datum();  i.prijave[i.brojac\_prijavljenih++] = novaP;  }  if (x == 5)  {  Ispit& i = odaberite\_ispit(ispiti, brojac\_ispita);  for (int j = 0; j<i.brojac\_prijavljenih; j++)  {  info(i.prijave[j].student);  cout << ", Unesite ocjenu sa ispita: ";  cin >> i.prijave[j].ocjena;  }  }  } while (x != 0);  } |