

LABORATORIUM 13. ZŁOŻONE TYPY DANYCH. STRUKTURY I UNIE.

Cel laboratorium:

Zaznajomienie z wykorzystaniem struktur i unii, jako typów złożonych do organizacji skomplikowanych danych. Nabycie praktycznych umiejętności wyboru dobrego sposobu reprezentacji danych oraz programowania z zastosowaniem struktur złożonych.

Zakres tematyczny zajęć:

- *pojęcie struktury,*
- *pojęcie unii,*
- *sposoby deklaracji struktur i unii,*
- *inicjalizacja struktur,*
- *dostęp do pól struktur i unii za pomocą zmiennej,*
- *dostęp do pól struktur i unii za pomocą wskaźników,*
- *typowe operacje na strukturach,*
- *strukтуры jako parametry funkcji.*

Kompendium wiedzy:

Pojęcie **struktura** można interpretować jako typ lub jako zmienną. Struktura, jako typ danych, składa się z pól i pozwala na przechowywanie danych różnych typów. Struktura, jako zmienna, to zmienna typu strukturalnego.

Deklaracja struktury: typ z nazwą + zmienna:

```
struct nazwaTypu {    //typ
    typPola1 nazwaPola1;
    typPola2 nazwaPola2;
    ...
};
struct nazwaTypu nazwazmiennej; //zmienna
```

```
struct student{
    char nazwisko[25];
    int wiek;
};
struct student st1;
```

Definicja struktury: deklaracja + inicjalizacja:

```
struct nazwaTypu zmienna={wartP1, wartP2,...};
struct student st2={"Kowalski",20};
```

Dostęp do składowych (pola struktury) za pomocą zmiennej:

```
zmienna.nazwaPola
```

```
st1.nazwisko
```



Dostęp do składowych (pola struktury) za pomocą wskaźnika do struktury:
<pre>struct nazwaTypu *wskaźnik; wskaźnik->nazwaPola //dostęp do pola</pre> <pre>struct student st3={"Kos",25}; struct student *wsk=&st3; printf("%d \n", st3.wiek); printf("%d \n", wsk->wiek); printf("%d \n", (*wsk).wiek);</pre> <p>Unia – typ złożony do przechowywania różnych rodzajów danych w tym samym obszarze pamięci (jednak nie równocześnie). Definicja unii jest analogiczna jak struktury tylko z innym słowem kluczowym:</p> <pre>union nazwaTypu { //typ typPola1 nazwaPola1; typPola2 nazwaPola2; ... }; union nazwaTypu nazwazmiennej; //zmienna</pre>
Dostęp do składowych (pola unii):
<pre>zmienna.nazwaPola //za pomocą zmiennej lub union nazwaTypu *wskaźnik;//deklaracja wskaźnika wskaźnik->nazwaPola //za pomocą wskaźnika</pre> <pre>union magazyn { int calkowita; double rzeczywista; char litera; }; union magazyn koszyk; koszyk.calkowita=22; //22 zapisane w koszyk, zajęte 2 bajty koszyk.rzeczywista=2.0; //22 usunięte, 2.0 zapisane, //zajęte 8 bajtów koszyk.litera='a'; // 2.0 usunięte, 'a' zapisane, zajęty 1 bajt</pre>

Pytania kontrolne:

1. Co to jest struktura?
2. Co to jest unia, czym różni się od struktury?
3. Jak odwołać się do pola struktury za pomocą zmiennej?
4. Jak odwołać się do pola struktury za pomocą wskaźnika?
5. Jak zainicjalizować pola struktury wartościami?
6. Jak wczytać i wyświetlić strukturę lub unię?
7. Czym różni się funkcja, w której parametrem jest struktura, od funkcji, w której parametrem jest wskaźnik na strukturę?



Zadania do analizy

Zadanie 13.1. Dane osobowe – przetwarzanie struktur

- Przeanalizuj przykład programu wykorzystującego struktury. Program wczytuje i wyświetla dane osobowe dwóch osób z wykorzystaniem funkcji z parametrem typu struktura lub wskaźnik na strukturę.
- Podaj tekst w komentarzach.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct daneos {                               //???
4      char imie[15];
5      char nazwisko[25];
6  };
7  struct daneos wczytaj();                       //???
8  void wyswietl(struct daneos os);               //???
9
10 void wczytaj2(struct daneos *wsk);             //???
11 void wyswietl2(struct daneos *wsk);            //???
12 //=====
13 int main(int argc, char *argv[])
14 { struct daneos osoba, osoba2;                 //???
15   printf("Osoba 1\n");
16   osoba=wczytaj();                             //???
17   wyswietl(osoba);                             //???
18   printf("Osoba 2\n");
19   wczytaj2(&osoba2);                           //???
20   wyswietl2(&osoba2);                          //???
21   system("PAUSE");
22   return 0;
23 }
24 //=====
25 struct daneos wczytaj()                       //???
26 { struct daneos os;
27   printf("Podaj imie "); gets(os.imie);         //???
28   printf("Podaj nazwisko "); gets(os.nazwisko);
29   return os;                                   //???
30 }
31 void wyswietl(struct daneos os)                 //???
32 {
33   printf("%s %s \n", os.imie, os.nazwisko); //???
34 }
35 void wczytaj2(struct daneos *wsk)               //???
36 { printf("Podaj imie ");
37   gets(wsk->imie);                             //???
38   printf("Podaj nazwisko ");
39   gets(wsk->nazwisko);
40 }
```



```
41 void wyswietl2(struct daneos *wsk)      //???
42 {
43     printf("%s %s \n",  wsk->imie,wsk->nazwisko);    //???
44 }
```

Zadanie 13.2. Dane klienta – przetwarzanie struktur i unii

- Przeanalizuj przykład programu wykorzystującego struktury i unie.
Program wczytuje i wyświetla dane osobowe dwóch klientów: jeden to osoba fizyczna, opisana imieniem, nazwiskiem i peselem, drugi to firma opisywana nazwą. Program wykorzystuje struktury i unie, pracę na zmiennych i na wskaźnikach.
- Podaj tekst w komentarzach.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct daneos {                      //???
4      char imie[15];
5      char nazwisko[25];
6      char pesel[11];
7  };
8  union dane {                         //???
9      struct daneos osoba;
10     char firma[60];
11 };
12 struct daneklienta {                 //???
13     int typ; //0-osoba,  inne-firma
14     union dane klient;
15 };
16 struct daneklienta wczytaj(int t);   //???
17 void wyswietl(struct daneklienta kl, int t); //???
18
19 void wczytaj2(struct daneklienta *wsk, int t); //???
20 void wyswietl2(struct daneklienta *wsk, int t); //???
21 //=====
22 int main(int argc, char *argv[])
23 { struct daneklienta klient1, klient2; //???
24   int wybor;
25   printf("Klient 1\n");
26   printf("Podaj typ klienta 0 - osoba,
27           inna wartość - firma\n");
28   scanf("%d", &wybor);                //???
29   klient1=wczytaj(wybor);              //???
30   wyswietl(klient1, wybor);            //???
31   printf("Klient 2\n");
32   printf("Podaj typ klienta 0 - osoba,
33           inna wartość - firma\n");
34   scanf("%d", &wybor);
```



```
35  wczytaj2(&klient2, wybor);           //???
36  wyswietl2(&klient2, wybor);          //???
37  system("PAUSE");
38  return 0;
39  }
40  //=====
41  struct dane klienta wczytaj(int t)     //???
42  { struct dane klienta temp;
43    if(t==0)
44      {printf("Podaj imie ");
45       scanf("%s",temp.klient.osoba.imie); //???
46       printf("Podaj nazwisko ");
47       scanf("%s",temp.klient.osoba.nazwisko);
48       printf("Podaj pesel ");
49       scanf("%s",temp.klient.osoba.pesel);
50       return temp;                     //???
51     }
52     else
53     {
54       printf("Podaj nazwe firmy ");
55       scanf("%s",temp.klient.firma);    //???
56       return temp;                     //???
57     }
58  }
59  void wyswietl(struct dane klienta kl, int t) //???
60  {
61    if(t==0)
62      {printf("%s %s %s \n", kl.klient.osoba.imie,
63       kl.klient.osoba.nazwisko, //???
64       kl.klient.osoba.pesel);} //???
65    else
66      {printf("%s\n", kl.klient.firma);} //???
67  }
68  void wczytaj2(struct dane klienta *wsk, int t)
69  { if(t==0)
70    { printf("Podaj imie ");
71     scanf("%s",wsk->klient.osoba.imie); //???
72     printf("Podaj nazwisko ");
73     scanf("%s",wsk->klient.osoba.nazwisko); //???
74     printf("Podaj pesel ");
75     scanf("%s",wsk->klient.osoba.pesel); //???
76    }
77    else
78    { printf("Podaj nazwe firmy ");
79     scanf("%s",wsk->klient.firma); //???
80    }
81  void wyswietl2(struct dane klienta *wsk, int t) //???
82  { if(t==0)
83    {printf("%s %s %s \n",
```



```
84     wsk->klient.osoba.imie,           // ???
85     wsk->klient.osoba.nazwisko,
86     wsk->klient.osoba.pesel);}
87     else
88     {printf("%s\n", wsk->klient.firma);}
89 }
```

Zadania do wykonania

Zadanie 13.3. Płaca pracownika

Oblicz płacę pracownika fizycznego. Zadeklaruj strukturę o polach: imię nazwisko, liczba godzin, stawka, premia w %.

Napisz funkcje do wczytywania danych, obliczania płacy i wyświetlania. Zastosuj przetwarzanie struktur przez zmienną i wskaźnik. Wywołaj te funkcje.

Zadanie 13.4. Płace grupy pracowników

Oblicz sumaryczną kwotę do wypłaty dla N pracowników fizycznych. Zadeklaruj tablicę struktur o polach: imię, nazwisko, liczba godzin, stawka, premia w procentach.

Wyświetl listę płac i sumaryczną kwotę pieniędzy do wypłaty.

Podaj nazwiska osób zarabiających najwięcej.

Zdefiniuj i wywołaj odpowiednie funkcje.

Zadanie 13.5. Nagrody dla pracowników

Zadeklaruj unię do przechowywania danych o nagrodach w postaci: albo kwota pieniężna, albo wycieczka w dane miejsce i w danym terminie albo list pochwalny. Dla N pracowników przypisz nagrody różnego typu.

Wyświetl przydział nagród w postaci: imię, nazwisko i dane nagrody.

Zdefiniuj i wywołaj odpowiednie funkcje.

Zadanie 13.6. Średnia ocen studenta i grupy

Zadeklaruj strukturę student o polach: imię, nazwisko, oceny (tablica 5 ocen). Dla grupy n studentów (tablica) oblicz średnią ocen każdego studenta i średnią grupy.

Zdefiniuj i wywołaj odpowiednie funkcje.

Zadania dodatkowe

Zadanie 13.7. Nagrody dla studentów

Zadeklaruj strukturę do przechowywania danych o szczęśliwych numerkach wylosowanych dla studentów (imię, nazwisko, numer). Dla grupy n studentów wpisz imiona i nazwiska, wylosuj im szczęśliwe numerki.

Ustal zwycięzców gry wg jednej z zasad:

- wygrywa osoba/osoby z największym numerkiem,
- wygrywa osoba/osoby, której numer jest najbliższy średniej wylosowanych liczb.

Zadanie 13.8. Cukierki

Zadeklaruj strukturę cukierek zawierającą informacje o nazwie producenta, nazwie cukierków, cenie za 1 kg, ilości w kg, dacie produkcji, dacie przydatności do spożycia. Wprowadź dane kilku takich struktur.

Wyświetl informacje o cukierkach wybranego producenta, których cena za 1 kg jest zawarta w przedziale od x do y zł.

Wyświetl informację o nazwach i ilości cukierków, którym skończył się termin przydatności do spożycia.

Zdefiniuj i wywołaj odpowiednie funkcje.

Zadanie 13.9. Plan zajęć

Zadeklaruj strukturę `pozycjaPlanu` zawierającą informację o zajęciach: dzień tygodnia, godzina rozpoczęcia, godzina zakończenia, przedmiot, wykładowca, nr sali. Wprowadź dane o swoim planie zajęć.

Wyświetl informacje o: zajęciach w danym dniu tygodnia, zajęciach z danego przedmiotu.

Podaj ile razy w tygodniu rozpoczynasz zajęcia o 8 rano.

Zdefiniuj i wywołaj odpowiednie funkcje.

Zadanie 13.10. Ankieta

Zadeklaruj strukturę `ankieta` zawierającą informacje: imię, nazwisko, adres (ulica, nr domu, nr mieszkania), płeć i wiek.

Określ:

- ile kobiet i ilu mężczyzn mieszka w jednym wybranym domu,
- ilu mężczyzn w wieku <18-60> lat mieszka na jednej wybranej ulicy.

Zdefiniuj i wywołaj odpowiednie funkcje.

Zadanie 13.11. Stypendium dla studentów

Napisz program wyznaczania stypendium studentom na podstawie wyników sesji zgodnie z zasadami:

- jeżeli wszystkie oceny to 5 – przyznawane jest podwyższone stypendium,
- jeżeli wszystkie oceny to 5 lub 4 – przyznawane jest zwykłe stypendium,
- jeżeli wśród ocen jest 3 – stypendium się nie należy.

Program powinien wyświetlać:

- listę studentów z ocenami i ich średnią,
- listę studentów z podwyższonym stypendium,
- listę studentów ze zwykłym stypendium.

Zdefiniuj odpowiednie funkcje dla pojedynczego studenta:

- `wczytywanieDanych`,
- `wyswietlanieDanych`,
- `przyznanieStypendium` z wynikiem: normalne/ podwyższone/ brak,
- `sredniaOcen`.

Zintegrowany Program Rozwoju Politechniki Lubelskiej – część druga

Zastosuj te funkcje do grupy n studentów, by wyświetlić ww. listy. Ustal założenia programu (oceny – liczby całkowite lub rzeczywiste, liczba ocen w sesji określana w programie lub stała).



Fundusze Europejskie
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz Społeczny

