

# Структури



# Определение

- съставен тип данни, включващ краен брой елементи, които може да са и от различен тип
- Елементите на структурата се наричат **полета**
- Достъпът до всяко поле се осъществява посредством името на структурата и името на полето
- Структурата е статичен тип данни

# Дефиниране на структура

```
struct име_тип  
{  
    тип1 име1;  
    ...  
    типN имеN;  
};
```

## □ пример:

```
struct точка  
{  
    int x;  
    int y;  
};
```

- **име\_тип** – име на структурата (име под което се дефинира нов тип данни)
- **тип1..типN** – типа на всяко от полетата
- **име1..имеN** – имената на полетата
  
- **Важно:**  
при дефинирането на структурата не се заделя никаква памет

# Деклариране на променлива от тип структура

име\_тип име;

□ може и така:

```
struct име_тип  
{  
    тип1 име1;  
    ...  
    типN имеN;  
} име;
```

име\_тип – име на структурата

име – име на променливата от тип структура

при декларирането на променлива от тип структурата се заделя толкова памет, колкото ще е необходима за всичките и елементи

# Инициализиране на структура

- става при декларацията:

```
struct tochka
{
    int x;
    int y;
};
```

```
tochka a = {4, 0};
tochka b = {6, -3};
```

- друг пример:

```
struct uchenik
{
    char ime[20];
    char prezime[20];
    char familia[20];
    int klas;
    char paralelka;
    double uspex;
};
```

- `uchenik x = {"Ivan", "Petkov", "Ivanov", 12, 'a', 5.70 };`

# Операции над цели структури

□ единствената допустима операция е присвояване

□ пример:

```
точка {  
    int x;  
    int y;  
};
```

```
точка a = {4, 0}, b;
```

```
b = a;
```

# Достъп до поле от структура

- достъпът става чрез **променлива.поле**
- поле от структура може да се използва навсякъде и да участва в операции както и обикновена променлива от неговия тип
- **пример:**
- `точка a, b;`
- `a.x = 10;`  
`a.y = -6;`  
`cin >> b.x >> b.y;`  
`if ((a.x==b.x)&&(a.y==b.y))`  
`cout << "Bingo!" << endl;`

# Пример - отпечатването на данни за ученик в подходящ вид

```
struct uchenik
{
    char ime[20];
    char prezime[20];
    char familia[20];
    int klas;
    char paralelka;
    double uspex;
};

uchenik x = {"Ivan", "Petkov",
            "Ivanov", 12, 'a', 5.70 };

cout << "Danni za uchenika:"
      << endl << endl;
cout << "Ime: "
      << x.ime << " "
      << x.prezime << " "
      << x.familia << endl;
cout << "Klas: " << x.klas
      << x.paralelka << endl;
cout << "Uspex: "
      << x.uspex << endl;
```



Край

