

Podstawy programowania

Struktury, tablice struktur

Struktury, wskaźniki do struktur

1. Przykład

Poniższy kod demonstruje podstawy pracy ze strukturami:

- **point** – klasyczna definicja typu strukturalnego,
- **vec** i **date** – struktury zdefiniowane z nazwą typu strukturalnego,
- **student** – typ strukturalny zagnieżdżony.

```
#include <stdio.h>
#include <string.h>
#include <math.h>

struct point
{
    int x;
    int y;
};

typedef struct vec
{
    double x;
    double y;
    double z;
} vec;

typedef struct date
{
    int day;
    int month;
    int year;
} date;

typedef struct student
{
    char name[50];
    char surname[50];
    int studentBook;
    date birthDate;
} student;

double vecLength(vec v)
{
    return sqrt(v.x * v.x + v.y * v.y + v.z * v.z);
}

int main( )
```

```
{
    struct point pa;
    pa.x = 10;
    pa.y = 20;
    printf("pa.x = %d\n", pa.x);
    printf("pa.y = %d\n\n", pa.y);

    struct point *pp = &pa;
    printf("pp->x = %d (strzałka)\n", pp->x);
    printf("( *pp ).y = %d (dereferencja)\n\n", (*pp).y);

    struct point pb;
    pb = pa;
    printf("pb.x = %d (kopia)\n", pb.x);
    printf("pb.y = %d (kopia)\n\n", pb.y);

    vec v = {3.0, 4.0, 0.0};
    double len = vecLength(v);
    printf("Długość wektora (%.2f, %.2f, %.2f) wynosi: %.2f.\n\n", v.x, v.y, v.z, len);

    student group[3];
    strcpy(group[0].name, "Wiktor");
    strcpy(group[0].surname, "Wektor");
    group[0].studentBook = 123456;

    group[0].birthDate.day = 15;
    group[0].birthDate.month = 11;
    group[0].birthDate.year = 2000;

    printf("Student: %s %s, nr indeksu: %d, ur.: %02d-%02d-%d.\n",
        group[0].name, group[0].surname, group[0].studentBook,
        group[0].birthDate.day, group[0].birthDate.month, group[0].birthDate.year);

    return 0;
}
```

- Przepisz i uruchom program.
- Zwróć uwagę na różne sposoby dostępu do pól struktury dla zmiennej automatycznej (z kropką .) i przy pomocy wskaźnika (strzałka ->).

2. Zadanie

- Napisz program do obliczania pól i obwodów figur płaskich (kwadrat i prostokąt). Wymiary figur powinny być przechowywane w strukturach. Program powinien zawierać:
- cztery funkcje do obliczania pól i obwodów z parametrem wskaźnika na strukturę,
- tablicę struktur jednego typu,
- funkcję obliczającą sumaryczne pole i obwody figur z tablicy korzystającą z wcześniej zdefiniowanych funkcji.

Zadania dodatkowe

1. Opracuj system reprezentacji samochodu ze strukturą zagnieżdżoną opisującą datę. Informacje o samochodach mają być przechowywane w tablicy rekordów.

```
typedef struct date
{
    int day;
    int month;
    int year;
} date;
```

```
typedef struct car
{
    char brand[30];
    char model[30];
    int year;
    date regDate;
} car;
```

Uzupełnij program o:

- funkcję drukującą informacje o samochodzie,
 - funkcję pozwalającą na dodanie samochodu przez użytkownika,
 - funkcję drukującą informacje o wszystkich samochodach.
2. Uzupełnij program o możliwość zapisu i odczytu tablic struktur do plików binarnych.