

Sam dla siebie C++  
każdy programować może

niedziela, dnia 19.10. 2025 r.

## *Struktury w C++ nr 008/25*

Struktura w języku C++ to pewna forma złożonego typu danych, która się odnosi do jakości i sposobu jakim są powiązane ze sobą elementy tworzące całość danego fragmentu programu. Bowiem program nie składa się tylko z kilku komend. Kilka komend tworzy mały program, o czym już wiadomość jako hobbysta zdążyłem nabyć. Tworząc program, należy potrzebne dane gdzieś zgromadzić. Ale bynajmniej tak zgromadzić, żeby nie zaśmiecać programu nieładem, tysiącami komend, powtarzanymi komendami, czy też nadając ciężar tak wielki, że komputer nie mógłby udźwignąć. Do uporządkowania tego rodzaju balastu programu, służą między innymi struktury w C++.

Struktura w C++ między innymi służy do:

- pogrupowania powiązanych danych;
- tworzenia obiektów z danymi i metodami;
- ułatwienia zarządzania kodem;
- zarządzania złożonymi danymi;
- tworzenia własnych typów danych, które są bardziej złożone niż typy wbudowane.

Definiowanie struktury w C++ rozpoczynam od słowa kluczowego **struct**, następnie podaję **unikalnaNazwa** i **nawiasy klamrowe** w których umieszczam deklarowane zmienne elementów mających swój tym i nazwę które będą należeć do tej struktury. Strukturę zakańcza się średnikiem. Brak średnika może spowodować nieprzeprowadzenie kompilacji.

**Oto sposoby definicji struktury:**

**Sposób nr 1.**

```
struct NazwaStruktury
{
    nazwaTypu1 nazwaZmiennych1
    nazwaTypu2 nazwaZmiennych2
    nazwaTypu3 nazwaZmiennych3
    .....
};
```

**Sposób nr 2.**

```
struct NazwaStruktury
{
    nazwaTypu1 nazwaZmiennych1
    nazwaTypu2 nazwaZmiennych2
    nazwaTypu3 nazwaZmiennych3
    .....
} nazwaZmiennej;
```

**Sposób nr 3.**

```

struct
{
    nazwaTypu1 nazwaZmiennych1
    nazwaTypu2 nazwaZmiennych2
    nazwaTypu3 nazwaZmiennych3
    .....
} nazwaZmiennej;

```

Sposób nr 3 jest raczej nie stosowany, a jeśli już to bardzo rzadko, bo brakuje nazwy dotyczącej struktury.

Nazwa struktury jest rozpoczynana dużą literą, chociaż to bywa niekonieczna, ale jak spostrzegłem tak przyjęto. Bowiem może to zapobiec popełnieniu błędów komputera w czasie kompilacji, gdyż przyjęty sposób stwarza różnicę pomiędzy nazwą struktury a nazwą zmiennych. Ilość elementów umieszczona w nawiasach klamrowych jest nieograniczona. Nie mniej należy zastosować logikę i umiar.

Mając na uwadze typy zmiennych, które są elementami struktury, spostrzegłem iż mogą to być typy dowolne, mogą być typy proste (np.: int, char, string, double, bool, ... ), oraz typy utworzone przez programistę tworzącego program, (mp.: typ wyliczeniowy, jakaś struktura, coś co było by potrzebne do programu.).

Struktura może zawierać następujące elementy, (cytat z internetu):

**zmienne składowe (pola danych):** Mogą to być zmienne dowolnego typu danych, w tym:  
 Typy podstawowe (np. int, double, char, bool).  
 Typy złożone (np. inne struktury, unie, klasy).  
 Wskaźniki (np. int\*  
 Tablice (np. char[20]).

**funkcje składowe (metody):** Struktury w C++ mogą zawierać metody, czyli funkcje operujące na polach danych struktury. Dzięki temu struktury mogą opisywać nie tylko stan, ale i zachowanie.

**specyfikatory dostępu:** W C++ struktury mogą używać specyfikatorów dostępu, takich jak public, private i protected, tak samo jak klasy. Domyślnie wszystkie elementy struktury są public.

**konstruktory i destruktory:** Struktury mogą mieć konstruktory do inicjalizowania swoich obiektów oraz destruktory do przeprowadzania operacji czyszczących.

**dziedziczenie:** Struktury w C++ mogą dziedziczyć po innych klasach i strukturach.

**szablony:** Podobnie jak klasy, struktury mogą być szablonowane (templatyzowane), co pozwala na tworzenie uogólnionych typów.

Oto prosty przykład struktury – Samochód:

```
/* pierwsza moja struktura – Samochód */  
  
#include <iostream>  
#include <cstdlib>  
  
// definicja struktury  
  
struct Samochod {  
    char marka[10];  
    char model[10];  
    int rok_produkcji;  
    double pojemnosc;  
    double przebieg;  
    double ladownosc;  
    char kolor[10];  
    char wlasciciel[15];  
};  
  
using namespace std;  
  
int main () {  
  
// utworzenie zmiennej typu samochod  
  
Samochod nieduzySamochod;  
  
/* tak wygląda przypisanie wartości do pól struktury za pomocą kropki - "."  
   oraz użycie strepy do kopiowania stringów do tablicy znaków */  
  
strepy(nieduzySamochod.marka, "Mercedes");  
strepy(nieduzySamochod.model, "ciezarowy");  
nieduzySamochod.rok_produkcji = 1994;  
nieduzySamochod.pojemnosc = 2.5;  
nieduzySamochod.przebieg = 9746.45;  
nieduzySamochod.ladownosc = 4.5;  
strepy(nieduzySamochod.kolor, "niebieski");  
strepy(nieduzySamochod.wlasciciel, "Pawel Pawleski");  
  
// wyświetlenie danych  
  
cout << "Marka: " << nieduzySamochod.marka << endl;  
cout << "Model: " << nieduzySamochod.model << endl;  
cout << "Rok produkcji: " << nieduzySamochod.rok_produkcji << endl;  
cout << "Pojemnosc: " << nieduzySamochod.pojemnosc << " L " << endl;  
cout << "Przebieg: " << nieduzySamochod.przebieg << " km " << endl;  
cout << "Ladownosc: " << nieduzySamochod.ladownosc << " ton " << endl;  
cout << "Kolor: " << nieduzySamochod.kolor << endl;  
cout << "Właściciel: " << nieduzySamochod.wlasciciel << endl;  
  
system("pause");  
return 0;  
}
```

A to wynik powyższego programu.

Marka: Mercedes  
Model: ciężarowy  
Rok produkcji: 1994  
Pojemność: 2.5L  
Przebieg: 9745.45km  
Ładowność: 4.5ton  
Kolor: niebieski  
Właściciel: Paweł Pawleski

*Tak dla przypomnienia – to jest moje hobby, programowaniem w C++ jestem zainteresowany i to sprawia mi przyjemność i zadowolenie, dlatego proszę nie posądzać mnie o plagiat, za opublikowanie wiadomości zacytowanych z internetu lub książki. Po prostu pokazuje z jakiego materiału korzystałem. I tyle. To nie jest forma zarobkowa.*

#### **Krótką uwaga.:**

Niniejsze opracowanie, jak i następne opracowania celem nauki własnej którą kontynuuję, opieram na lekturach takich jak:

1. Język C++ Szkoła programowania Stephen Prata
2. Programowanie w języku C++ Wiesław Porębski
3. Podstawy języka C++ Stanley B. Lippman
4. Symfonia C++ standard Jerzy Grębosz
5. PRZEWODNIK DLA POCZĄTKUJĄCYCH C++ ALEX ALLAIN
6. Zadania z programowania z przykładowymi rozwiązaniami Mirosław J. Kubiak
7. i internet

#### **Ćwiczenie nr 1.**

Mając już za sobą pokonany pewny trud w nabywaniu wiedzy o programowaniu w C++ , warto spróbować coś utworzyć, co by było satysfakcją z poświęconego czasu i nabytej umiejętności.

Dlatego też, utworzyłem sobie przykładową strukturę, w której są zapisywane dane kierowcy firmy fikcyjnej „Jakieś Tam” usługowej trzy osobowej.

Nim utworzyłem program struktura, na kartce z papieru zapisałem co chcę aby było w tym programie C++.

Oto dane:

- nazwisko
- imię
- pesel
- nr dowodu osobistego
- płeć
- pojazd do zarobkowania
- marka pojazdu
- model pojazdu

- rok produkcji pojazdu
- numer rejestracyjny
- numer boczny firmowy
- kolor pojazdu.

Wyżej podane elementy, są składowymi programu w C++ , to jest struktura informacyjna rejestru kierowców.

Oto utworzony program, tak wygląda na plik źródłowy:

```

/* firma taxi „Jakaś Tam”- dane kierowcy */

#include <iostream>
#include <cstdlib>

using namespace std;

struct dane_taksowkarza
{
    char nazwisko[30];
    char imie[20];
    int pesel;
    char nr_dowodu[15];
    char plec[20];
    char pojazd[30];
    char marka[30];
    char model[30];
    int rok_produkcji[30];
    char nr_rejestracyjny[30];
    int nr_firmowy[30];
    char kolor[30];
}

int main ()
{
    dane_taksowkarza kierowca[3];
    int i;

    cout << "Firma Jakaś Tam rejestr kierowcow." << endl;
    cout << endl;

    for ( i = 1; i <= 3; i++)
    {
        cout << "Podaj nazwisko: " << i + 1 << ".kierowcy" << endl;
        cin >> kierowca[i].nazwisko;

        cout << "Podaj swoje imie: " << i + 1 << ".kierowcy" << endl;
        cin >> kierowca[i].imie;

        cout << "Podaj pesel: " << i + 1 << ".kierowcy" << endl;
        cin >> kierowca[i].pesel;

        cout << "Podaj numer dowodu osobistego: " << i + 1 << ".kierowcy" << endl;
        cin >> kierowca[i].nr_dowodu;
    }
}

```

```

cout << "Podaj swoja plec: " << i + 1 << ".kierowcy" << endl;
cin >> kierowca[i].plec;

cout << "Podaj jakim pojazdem zarobkujesz: " << i + 1 << ".kierowcy" << endl;
cin >> kierowca[i].pojazd;

cout << "Podaj marke pojazdu: " << i + 1 << ".kierowcy" << endl;
cin >> kierowca[i].pojazd;

cout << "Podaj model pojazdu: " << i + 1 << ".kierowcy" << endl;
cin >> kierowca[i].model;

cout << "Podaj rok produkcji pojazdu: " << i + 1 << ".kierowcy" << endl;
cin >> kierowca[i].rok_produkcji;

cout << "Podaj numer rejestracyjny pojazdu: " << i + 1 << ".kierowcy" << endl;
cin >> kierowca[i].nr_rejestracyjny;

cout << "Podaj numer firmowy pojazdu: " << i + 1 << ".kierowcy" << endl;
cin >> kierowca[i].nr_firmowy;

cout << "Podaj kolor: " << i + 1 << ".kierowcy" << endl;
cin >> kierowca[i].kolor;

cout << endl;
cout << "===== " << endl;
cout << " Podane informacje o kierowcy " << endl;
cout << "===== " << endl;

for ( i = 0; i <=3; i++ )

cout << "Informacje o taksówkarzu: " << i + 1 << ".kierowcy" << endl;

cout << "Nazwisko: " << kierowca[i].nazwisko << endl;
cout << "Imie: " << kierowca[i].imie << endl;
cout << "Pesel: " << kierowca[i].pesel << endl;
cout << "Numer dowodu osobistego: " << kierowca[i].nr_dowodu << endl;
cout << "Plec: " << kierowca[i].plec << endl;
cout << "Pojazd do zarobkowania to: " << kierowca[i].pojazd << endl;
cout << "Marka pojazdu: " << kierowca[i].marka << endl;
cout << "Model pojazdu" << kierowca[i].model << endl;
cout << "Rok produkcji pojazdu: " << kierowca[i].rok_produkcji << endl;
cout << "Numer rejestracyjny pojazdu: " << kierowca[i].nr_rejestracyjny << endl;
cout << "Numer firmowy pojazdu: " << kierowca[i].nr_firmowy << endl;
cout << "Kolor pojazdu: " << kierowca[i].kolor << endl;
}
cout << "===== " << endl;

system ("pause");
return 0;
}

```

Efekt działania programu

Firma Jakaś Tam rejestr kierowcow.

Podaj nazwisko: 1.kierowcy

—

No i z klawiatury podajesz o co prosi program na komputerze.

*Fajne. Prawda?  
Też tak myślę że fajne.*

*No i super.*