**Struktury w C++**

W pewnych sytuacjach, typy zmiennych, które oferuje język C++ mogą być niewystarczające. Mamy do dyspozycji szereg typów prostych takich jak **int, char, bool**czy typy złożone zwane **tablicami**. Tablica to taka struktura danych, która przechowuje wiele elementów **tego samego typu.** Co jeśli potrzebujemy pracować na danych, których składowe są różnych typów? Jest tak np. w bazie danych. Na ratunek przychodzą nam **struktury,** czyli złożone typy danych, które definiuje programista.

***Tworzenie struktury*** - struktury tworzymy za pomocą słowa kluczowego **struct** podając jej nazwę (tak będzie nazywał się typ przez nas stworzony), a następnie w nawiasie klamrowym definiujemy jej elementy:

struct nazwa

{

typ nazwa\_elementu;

typ nazwa\_drugiego\_elementu;

typ nazwa\_trzeciego\_elementu;

...

};

***Przykład***

struct samochod

{

string marka; //pola struktury

string model;

int rok\_produkcji;

double pojemnosc;

};

***Tworzenie zmiennych typu strukturalnego*** - po stworzeniu struktury mamy nowy typ - w naszym przypadku będzie nazywał się on **samochod**. Możemy teraz tworzyć zmienne na jego podstawie. Nie różni się ono niczym od tworzenia zmiennych opartych na typach wbudowanych takich jak **int, char** itp. Prześledźmy przykład:

int main()

{

int a; //tworząc zmienną podajemy jej typ i nazwę

samochod renault; //w tym przypadku jest tak samo, ponieważ

//samochod jest typem, a renault nazwą

  system("pause");

return 0;

}

***Odwoływanie się do pól składowych*** - do pól odwołujemy się za pomocą kropki, podając najpierw nazwę zmiennej strukturalnej, a następnie po kropce nazwę pola struktury:

int main(){

  samochod renault;

  cin>>renault.marka; //odwołanie się do pola marka

cin>>renault.pojemnosc; //odwołanie się do pola pojemnosc

  cout<<"Marka: "<<renault.marka<<", pojemnosc:"<<renault.pojemnosc<<endl;

system("pause");

return 0;

}

***Nadanie wartości początkowych*** - wartości początkowe nadajemy podobnie jak w tablicach. Taki sposób można zastosować tylko przy inicjowaniu zmiennej strukturalnej. Wartości pól podajemy w nawiasie klamrowym pamiętając o kolejności przypisywanych elementów. Jeśli pierwszym polem jest ciąg znaków, to taki ciąg powinien znaleźć się jako pierwszy. Dotyczy to wszystkich pól składowych. Rozpatrzmy przykład:

int main()

{

//nadanie wartości początkowych

samochod renault = {"renault","megane",2013,2.0};

//wypisanie wartości elementów struktury

cout<<renault.marka<<endl<<renault.model<<endl<<renault.rok\_produkcji

<<endl<<renault.pojemnosc<<endl;

system("pause");

return 0;

}

Drugim sposobem jest przypisanie wartości od razu przy tworzenie struktury:

struct samochod{

string marka;

string model;

int rok\_produkcji;

double pojemnosc;

} renault = {"renault","megane",2013,2.0};

[***Tablice struktur***](http://www.algorytm.edu.pl/struktury-unie-pola-bitowe-wyliczenia/tablice-struktur.html) - tablicę struktur tworzymy i odwołujemy się do niej w ten sam sposób co [zwykłe tablice prostych zmiennych](http://www.algorytm.edu.pl/tablice-w-c/tablice-jednowymiarowe-w-c.html).

***Przykład***

#include<iostream>

#include<cstdlib>

using namespace std;

struct punkty{

int x, y;

char nazwa;

};

int main()

{

punkty tab[1000];

  //przypisanie wartosci do pierwszej komórki tablicy

tab[0].x = 2;

tab[0].y = 4;

tab[0].nazwa = 'A';

  //przypisanie do ostatniej komórki tablicy

tab[999].x = 1;

tab[999].y = 5;

tab[999].nazwa = 'X';

  //odwoływanie się do elementów tablicy

cout<<"Dane pierwszego punktu: "<<tab[0].x<<" "

<<tab[0].y<<" "<<tab[0].nazwa<<endl;

system("pause");

  return 0;

}

***Ćwiczenie 1*** Farmer zwrócił się do Ciebie z prośbą byś napisał program, który będzie przechowywał informacje o rodzajach zwierząt jakie posiada. Zwierzęta to krowa, koza, kura, pies i świnia. Cechy tych zwierząt to nazwa, waga, wiek (podany w tygodniach), oraz data nabycia (zakupu). Farmer dodał, iż krowę i świnię kupił 24.04.14r. od znajomego, pies przybłąkał się 6.05.13r., natomiast kozę i kurę dostał w prezencie od wójta 23.02.15r. Niestety resztę cech musisz określić sam, na podstawie przeciętnych dla tych zwierząt. Dane mają być zawarte w strukturach.

1. Zrealizuj wyświetlenie wszystkich danych w przejrzysty i zrozumiały dla użytkownika sposób, użytkownik nic nie podaje, program wyświetla tylko dane.
2. Następnie napisz funkcje:

* wypisz – robi to samo co w a),
* zakup – dopisuje kolejne zwierzę farmerowi do inwentarza,
* sprzedaz – (dla chętnych) usuwa zwierzę z inwentarza.