POLITECHNIKA ŚWIĘTOKRZYSKA
Wydział Elektrotechniki, Automatyki i Informatyki
Katedra Elektrotechniki Przemysłowej i Automatyki

Programowanie komputerów 2

Typy złożone

Definiowanie typów pochodnych, typ wyliczeniowy, struktury

Paweł Strączyński

2020

1 Typy pochodne - typedef

Słowo kluczowe **typedef** służy do definiowania nowego typu w oparciu o te istniejące. Przykład wykorzystania słowa kluczowego **typedef** przedstawiono poniżej.

Listing 1.1: Tworzenie typu pochodnego

```
1 typedef unsigned int  uint32_t;
2 typedef char int8_t;
3 typedef int* p_int;
```

2 Typ wyliczeniowy - enum

Typ wyliczeniowy **enum** służy do do tworzenia takich zmiennych które mogą przyjmować tylko określone wartości. Definicja typu wyliczeniowego składa się z:

- słowa kluczowego **enum**,
- nazwy typu,
- listy nazw symbolicznych ujętych w nawiasy klamrowe.

Przykład tworzenia zmiennych typu wyliczeniowego przedstawiono na listingu poniżej.

Listing 2.2: Tworzenie typu pochodnego

```
1 //definicje typu wyliczeniowego
2 enum Sekwencja {Ap, Bp, Am, Bm}
3 enum Kierunek {LEWO, PRAWO}
4
5 //utworzenie zmiennych typu wyliczeniowego
6 enum Sekwencja ruch = Ap;
7 enum Kierunek kier_obr = LEWO;
```

Użycie zmiennych z listingu 2.2 przedstwiono ponizej.

Listing 2.3: Tworzenie typu pochodnego

```
1 switch(ruch)
2  {
3    case Ap:
4      AP;
5      ruch = (kier_obr == PRAWO ) ? Bp : Bm;
6      break;
```

```
7     case Bp:
8       BP;
9       ruch = (kier_obr == PRAWO ) ? Am : Ap;
10      break;
11    case Am:
12      AM;
13      ruch = (kier_obr == PRAWO ) ? Bm : Bp;
14      break;
15    case Bm:
16      BM;
17      ruch = (kier_obr == PRAWO ) ? Ap : Am;
18      break;
19  }
```

3 Struktury

Struktury są specjalnym typem danych w języku C który może przechować wiele wartości różnego typu w jednej zmiennej. Poniżej przedstawiono przykładową definicję struktury:

Listing 3.4: Przykład definiowania struktury oraz tworzenia zmiennej typu strukturalnego

```
1 struct Samochod { //Definicja typu strukturalnego
2     char* marka;
3     char* model;
4     int rok_produkcji;
5     int vin;
6  };
7
8 struct Samochod samochod1; //Utworzenie zmiennej typu strukturalnego
```

Dostęp do poszczególnych pól struktury możliwy jest z wykorzystaniem operatora **.** (kropki) i został przedstawiony na listingu 3.5.

Listing 3.5: Dostęp do pól struktury

```
1 //Wpisywanie danych do struktury
2 samochod1.marka = "Tesla";
3 samochod1.model = "S";
4 samochod1.rok_produkcji = 2019;
5
6 //Drukowanie danych struktury
7 printf("SAMOCH D␣1:\n");
8 printf("␣␣␣␣MARKA:␣%s\n",samochod1.marka);
```

```
9  printf("␣␣␣␣MODEL:␣%s\n",samochod1.model);
10 printf("␣␣␣␣ROK␣PRODUKCJI:␣%s\n",samochod1.rok_produkcji);
```

Nowa strukturę można wypełnić danymi od razu podczas jej tworzenia:

Listing 3.6: Inicjalizacja pół struktury

```
1 struct Samochod samochod2 = {'Nissan', 'Leaf', 2020};
```

Na strukturę tak samo jak na każdą inną zmienną może wskazywać wskaźnik. Przykład użycia przedstawiono poniżej.

Listing 3.7: Wskaźnik na strukture

```
1  typedef struct {
2     char* marka;
3     char* model;
4     int rok_produkcji;
5     int vin;
6   } Samochod;
7
8   int main ()
9   {
10    Samochod samochod1 = { 'Tesla','S',2019 };
11    Samochod *wsk = &samochod1;
12    wsk->model = 'S';
13    return 0;
14  }
```

Zapis `wsk->model` jest z definicji równoważny zapisowi `(*wsk).model`. Zapis ten ze wzgleu na przejrzystość jest powszechnie stosowany. Wyrażenie `wsk.model` spowoduje błąd kompilacji ponieważ strukturą jest `*wsk` a nie `wsk`.

4 Zadania do wykonania:

1. Napisz program implementujący bibliotekę książek w postaci n elementowej tablicy struktur. Program powinien pobierać dane ze standardowego wejścia i drukować na standardowe wyjście. Struktura opisująca książkę powinna zawierać pola:

   - tytuł,
   - autor,
   - rok_wydania,

- status.

Pole status zdefiniować jako zmieną typu wyliczeniowego. Do odczytu ciągów znaków zapisywanych w polach "tytuł" i "autor" użyć funkcji `fgets`. Dane zapisać w buforze pomocniczym. W celu usunięcia znaku końca linii z odczytanego ciągu znaków użyć funkcji `sscanf`. Przed odczytem kolejnych danych wyczyścić bufor pomocniczy przy pomocy funkcji `memset`. Do odczytu wartości numerycznej użyć funkcji `scanf`.

2. Zmodyfikować program z zadania 1 tak aby drukowanie danych z tablicy struktur realizowane było za pomocą wskaźnika na kolejną strukturę w tablicy.

3. Zmodyfikować program z zadania 2 tak aby operacje wczytywania danych do struktury (NIE CAŁEJ TABLICY) i ich drukowania realizowane były w osobnych funkcjach. Funkcje powinny pobierać wskaźnik do danej struktury jako argument wejściowy.