

# Složené datové typy

## - Výčtový datový typ

```
enum weekday {MON, TUE, WED};  
enum weekday day;
```

- 
- Do proměnné day se zapíše jako kdyby index 0 z weekday

```
int iday = MON;  
//iday = 0
```

- 

```
weekday {..., SUN, N_DAYS}
```

- 
- Počet dní se dá zapsat jako poslední objekt ve weekday

## - Datová struktura

```
typedef struct {  
    char *login;  
    int subjects[3];  
    int n_subjects;  
} student;
```

```
student user;
```

- 
- Přístupování k položkám

```
user.n_subjects = 10;  
user.login = "Login";  
//user.login[0] = 'L'
```

- 
- Při použití s pointerem

```
student *p;  
p = &user;  
// můžu použít (*p).n_subjets, ale spíš se používá  
p -> n_subjects
```

- 
- Inicializace

- Můžu udělat tohle:

```
user.login = "xlogin";  
user.n_subjects = 0;
```

- Ale je efektivnější si to extrahovat do funkce, např. create\_student(student \*user)

- Přidávání předmětů

```
enum subjects {IZP, IUS};  
user.subjects[user.n_subjects-1] = IZP;  
user.n_subjects++;
```

- Taky se to může extrahovat do funkce, např. add\_subject(IZP, student \*user)