

### Funcții polimorfice.

O *entitate polimorfică* poate avea mai multe tipuri (polimorfism = mai multe forme). O funcție polimorfică poate fi apelată cu parametri de tip diferit. La definirea unei funcții polimorfice se vor folosi parametri de tip **void \*** (pointeri generici). Aceștia vor fi înlocuiți, la apelul funcției, prin pointeri la tipuri definite.

În cazul pointerilor generici apar dificultăți legate de:

- accesul la informația referită printr-un pointer generic – pentru a accesa informația vom utiliza un pointer cu tip (de exemplu **char\***)
- compararea informației referite prin pointeri generici – funcția polimorfică va conține un pointer la o funcție de comparație de informații referite prin **void\***.

Aceasta va fi înlocuită printr-o funcție de comparație definită de utilizator.

Exemplificăm prin definirea unei funcții polimorfice de căutare binară.

Informația indicată prin pointerii **void \*** o accesăm prin pointeri la **char\***.

```
//definire tip pointer la functie de comparatie generală
typedef int (*PFCP)(void* ch, void* el);

//functie polimorfica de cautare binara
void* CautBin(void* ch, void* tb,
              int n, int dim, PFCP fcp){
    // ch=pointer la cheia cautata
    // tb=pointer la tabelul in care se face cautarea
    // n=numarul de elemente din tabel
    //dim=numarul de octeti al unui element din tabel
    //fcp=pointer la functia generala de comparatie
    int rezcomp;
    //pointer la primul element din tabel
    char *min = (char*) tb;
    char *med;
    //pointer la ultimul element din tabel
    char *max = (char*) tb + (n-1)*dim;
    while(min <= max){
        med = min + (max-min)/dim/2*dim;
        rezcomp = (*fcp)(ch, med);
        if(rezcomp < 0)
            max = med - dim;
        else
            if(rezcomp > 0)
                min = med + dim;
            else
                return med;
    };
    return NULL;
}
```

Pointerii la elementele de același tip pot fi comparați (**min** <= **max**). Pointerul **med** se obține pe o cale mai ocolită: se adună la pointerul **min** o valoare constantă – distanța între **min** și **max**, exprimată în octeți (**max-min**) este convertită mai întâi în indice, prin împărțirea cu dimensiunea elementului **dim**, este apoi împărțită la 2, pentru a obține mijlocul și transformată din nou în octeți.

Funcția de comparație are ca parametri pointeri generici la valorile comparate și întoarce o valoare negativă, 0 sau o valoare pozitivă, în funcție de rezultatul comparației. Ea va fi adresată printr-un pointer (pointer la funcție) .

Explicitarea funcției concrete de comparație se va face în apel. Vom considera două cazuri: compararea a două valori întregi, respectiv compararea a două șiruri de caractere. Pointerii generici sunt convertiți în pointeri la tipul respectiv (întreg sau șir de caractere), se face dereferențierea și se returnează ca rezultat diferența valorilor, respectiv rezultatul comparației șirurilor de caractere.

```
int cpint(void* ch, void* el){  
    return *(int*)ch - *(int*)el;  
};
```

```
int cpsir(void* ch, void* el){  
    return strcmp((char*) ch, (char*)el);  
};
```