Estruturas de Dados Registros

Um registro é um tipo de dado construído, definido por meio de algum mecanismo específico da linguagem, que é composto por vários itens de dados que são armazenados juntos e tratados como uma unidade. Cada item de dado componente de um registro tem um nome próprio e pode ser de um tipo distinto dos demais itens de dados daquele registro e por esse motivo dizemos então que os registros são arranjos heterogêneos de dados.

Em linguagem C os registros são chamados de <u>structs</u> e cada item de dado de uma <u>struct</u> é chamado de <u>membro</u> daquela <u>struct</u>. Um outro termo utilizado para designar os membros de uma <u>struct</u> é 'campo'.

Exemplos comuns de utilização prática:

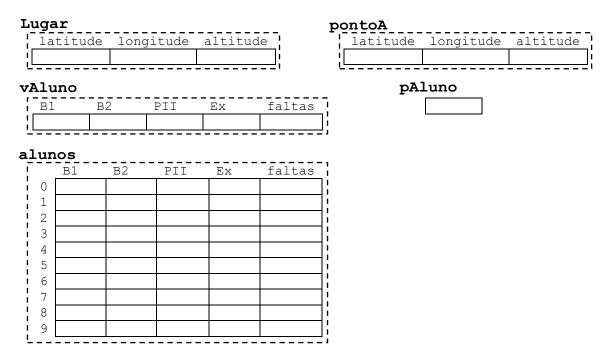
- coordenadas geográficas de um ponto;
- conjunto de notas e faltas de um aluno;
- dados cadastrais de um cliente.

O programa a seguir ilustra a declaração e manipulação de registros em linguagem C.

```
#include <stdio.h>
/* Definindo uma struct para armazenar pontos geográficos */
struct pontoGeogr { float latitude; float longitude; float altitude;};
/* Definindo uma struct para armazenar os resultados de um aluno */
struct resultAluno { float B1; float B2; float PII; float Ex; int faltas;};
/* Atribuindo um sinônimo para a struct pontoGeogr */
typedef struct pontoGeogr PGeogr;
int main(void)
                                   /* criando a variavel Lugar */
     struct pontoGeogr Lugar;
                                   /* variável pontoA é do mesmo tipo que a
     PGeogr pontoA;
                                      variável Lugar */
     struct resultAluno vAluno;
     struct resultAluno *pAluno; /* ponteiro para uma struct resultAluno */
     struct resultAluno alunos[10];
                                      /* vetor de struct */
     pAluno = &vAluno;
     /* atribuindo valores para os membro da variável vAluno */
     vAluno.B1 = 4.5;
     vAluno.B2 = 6.0;
     vAluno.PII = 2.0;
     vAluno.Ex = 7.5;
     vAluno.faltas = 0;
     /* atribuindo para o membro faltas da struct através do ponteiro */
     pAluno->faltas = 5;
      /* atribuindo para o membro faltas de um elemento do vetor de struct */
     alunos[5].faltas = 2;
     return 0;
}
```

Estruturas de Dados Registros

Esquematicamente podemos representar o estado inicial das 5 variáveis definidas no programa anterior da seguinte forma:



Ao final do programa, com as atribuições que foram feitas, o estado das variáveis seria:

