ESTRUTURA DE DADOS

APRESENTANDO AS ESTRUTURAS DE DADOS

Olá!

Ao final desta aula, você será capaz de:

1. Compreender o conceito de Estruturas de Dados;

2. Compreender o conceito de Função;

3. Compreender o conceito de struct;

4. Aplicar os conceitos de Árvore, Grafo, Pilha, Fila e Lista.

1 Tema: Da nascente do Rio Ailã no Monte Caburaí (RR) ao Arroio Chuí(RS) - uma viagem para conhecer as Estruturas de Dados e outros tópicos

Ao iniciarmos o estudo de Algoritmos, tudo pareceu tão abstrato e se fossemos realizar uma eleição, com certeza, o conceito de variável ganharia.

Como era difícil entender que a variável era um lugar na Memória Principal e que esse lugar não era um, às vezes era mais do que um. Complicado não era?

Além disso, você tinha que nomear e associar um valor a ele. E o mais difícil era entender que, em um momento, o nome representava o lugar e, em outro momento, o valor.

É por essa e muitas outras razões que a Ciência da Computação ficou conhecida como a Ciência da Abstração.

Mas, apesar de tudo isso, com o passar do tempo, você foi incorporando o conceito e usando os tipos int, float, char e double sem nem mesmo saber como os valores eram armazenados na memória e como foram implementadas as operações. Simplesmente usou.

Agora, chegou sua vez de criar **Tipos Abstratos de Dados** (TADs), listar suas operações e implementá-las na linguagem C++ para torná-los "concretos".

O conteúdo é muito extenso e os conceitos muito abstratos. Sendo assim, o nosso maior desafio será: como tornar essa disciplina amigável e apaixonante porque Algoritmos e Estruturas de Dados são elementos indispensáveis para você se tornar um desenvolvedor.

2 Viajando para se familiarizar com os termos

"Estrutura de Dados é a organização e a representação das informações, entre outras, na forma de pilhas, filas, árvores, listas ligadas e vetores, geralmente na memória do computador para obter a devida abstração de um problema real e a melhor eficiência na execução dos algoritmos cujas operações atuam sobre essas estruturas de dados." (MORAES. C.R., 2001, p. 9)

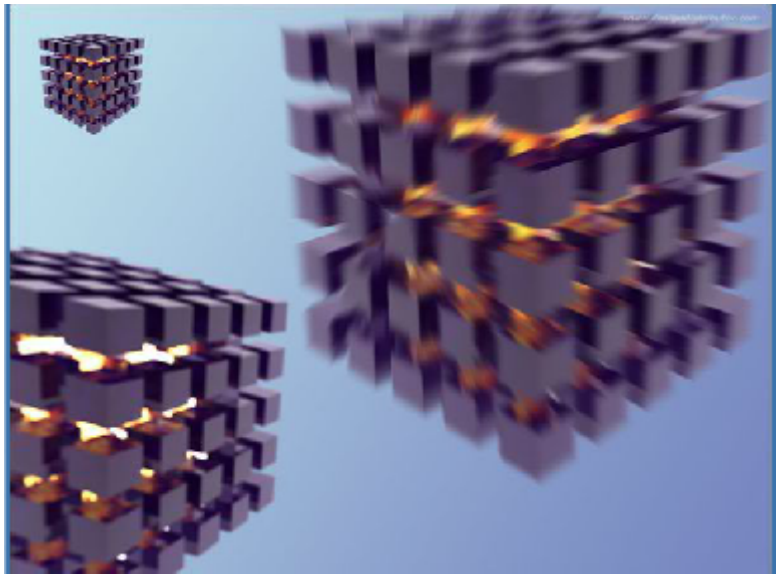
Conceituar Estruturas de Dados não é unia tarefa fácil porque não se pode ser prolixo nem tão pouco conciso. Afinal, são muitos detalhes a serem explicados. Então, só nos resta conhecê-las para que possamos, ao final do curso, construirmos nosso conceito sobre as Estruturas de Dados.

Urna das classificações sobre as Estruturas de Dados diz respeito à Alocação na Memória, um questionamento que alguns de vocês fizeram na disciplina de Algoritmos: **"Não dá para determinar o tamanho do vetor na hora da execução?"**

Vocês trabalharam com Alocação Estática na disciplina de Algoritmos e, até a sétima aula de Estrutura de Dados, continuarão trabalhando. Na AULA 8, irão estudar Alocação Dinâmica e, com certeza, algumas das suas dúvidas serão respondidas.

Gostaríamos que você entendesse a importância de conhecer profundamente as Estruturas de Dados, mas, para que isso seja possível, precisa entender como as informações são armazenadas na memória, construir algoritmos com facilidade para poder manipular as informações que se encontram dentro das Estruturas de Dados e, o mais importante, ser capaz de escolher a estrutura certa para cada aplicação. (PREISS, B., 2000, p.1)

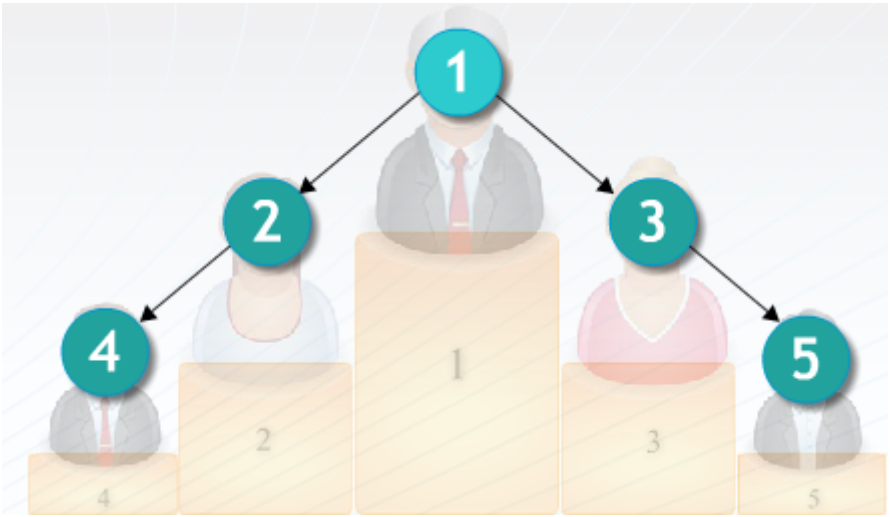
Depois de tanta abstração, para começarmos de forma mais "light" nosso estudo, gostaríamos que você lesse essa história com muita atenção e retornasse toda vez que sentisse necessidade.



Sr. Augusto, viúvo, tem duas filhas e cada uma das suas filhas tem um filho. Poderíamos representar a família atual do Sr. Augusto da seguinte maneira:

A figura representa uma **árvore** *binária* porque tem duas *subárvores* e para ser considerada uma árvore binária, tem que ter, no máximo, grau dois.

O *grau* de uma **árvore** é definido pelo número de subárvores de um nó. Sendo assim, os nós 2 e 3 têm grau 1 e os nós 4 e 5, grau 0, por essa razão, são chamados de *terminais* ou *folhas*. As aplicações são muitas, pois tudo que requer hierarquia, pode usar árvore. Um organograma, por exemplo.



Neste ano, Sr. Augusto resolveu levá-los para viajar nas férias. Um dia, os chamou para se encontrarem e apresentar o roteiro. Quando todos chegaram, ele disse que tinha escolhido uma viagem muita longa, mas interessante, pois no tempo de escola dele, o ponto extremo norte era um e agora ele tinha lido que havia mudado. Todos olharam espantados. Será que eles iriam atravessar o Brasil de Norte a Sul?

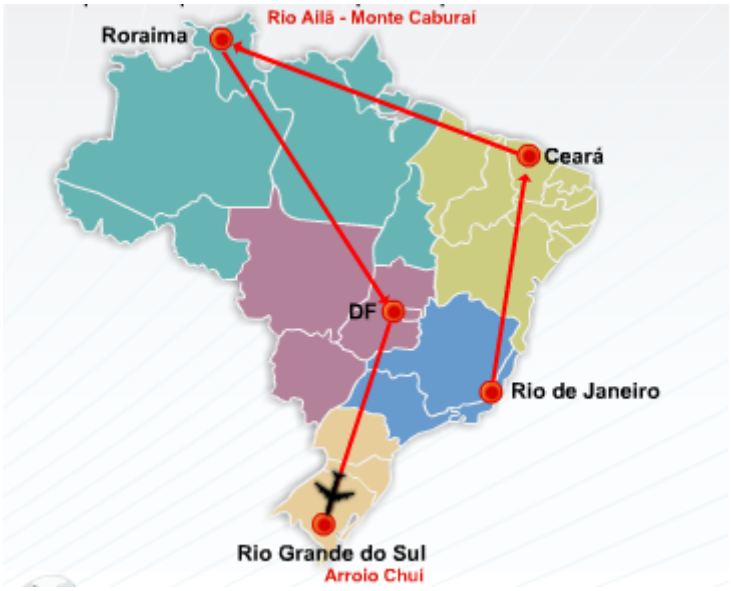
Foi aí que ele apresentou o mapa e depois o roteiro como veremos a seguir:

Mapa com os pontos de vista.

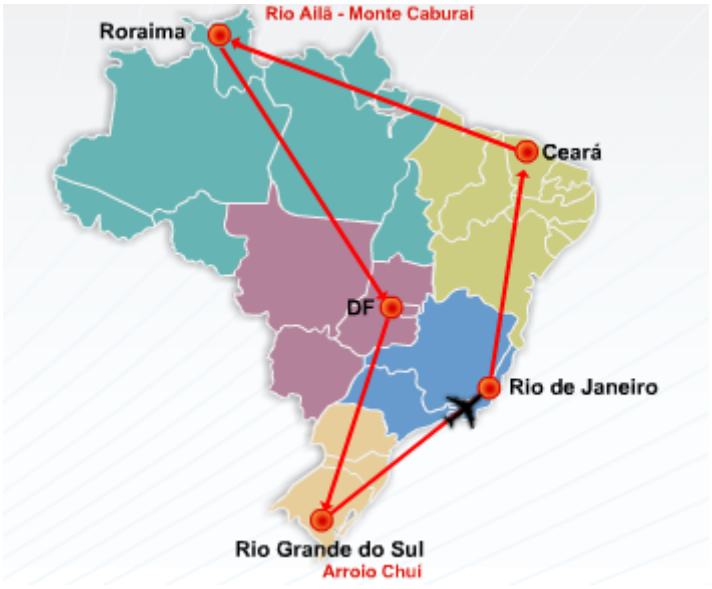
Saída do ponto de partida e chegada no primeiro ponto de destino.

Saída do primeiro ponto de destino e chegada no segundo ponto.

Saída do segundo ponto e chegada no terceiro ponto.



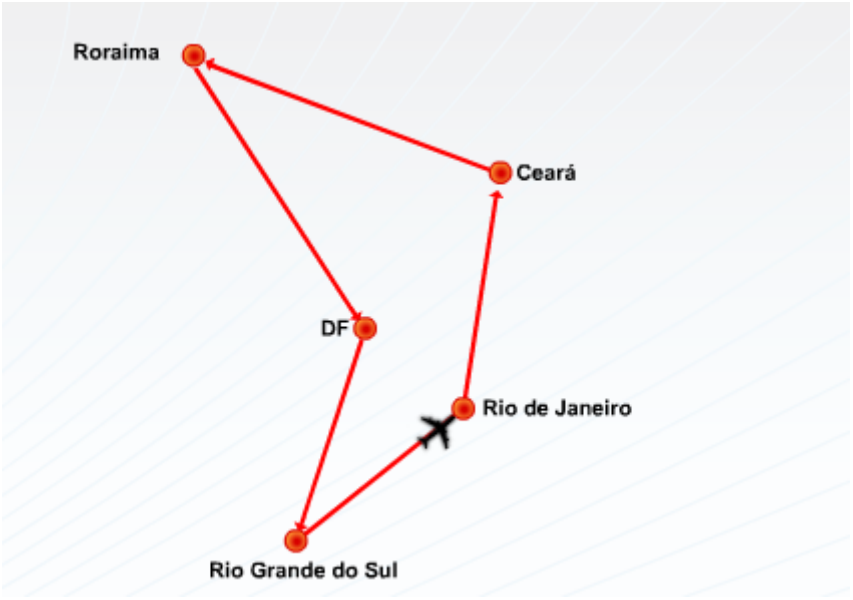
Saída do terceiro ponto e chegada no quarto ponto.



Saída quarto ponto e retorno ao ponto de origem.

Realmente eles iriam atravessar o Brasil. Claro que não foi o menor caminho, mas o melhor voo era o que fazia conexão em Fortaleza e assim aproveitaram para conhecer a terra de Iracema de José de Alencar para depois seguirem viagem para Roraima.

O roteiro da viagem está representado na figura abaixo.



Grafo

O grafo é uma Estrutura de Dados não linear. Podemos defini-la como sendo um conjunto de nós (vértices) e suas conexões (arcos) entre eles. Além disso, não existe limitações para os vértices.

Embora não seja objeto de estudo dessa disciplina, precisamos saber que são usados em muitas aplicações como as relacionadas abaixo:

- **Redes de computadores;**
- **Computação Gráfica;**
- **Diagrama de Entidade Relacionamento (E-R);**
- **Modelagem de Circuitos Digitais, entre outras.**

No dia da viagem, eles estavam tão ansiosos que chegaram bem cedo ao aeroporto Tom Jobim no Rio de Janeiro e, quando o *check in* abriu, foram os primeiros da **fila.**

A **fila** é uma estrutura de dados muito usada em computação. A Fila segue a regra chamada: **FIFO** - *first in, first out* (o primeiro elemento que entra é o primeiro que sai).

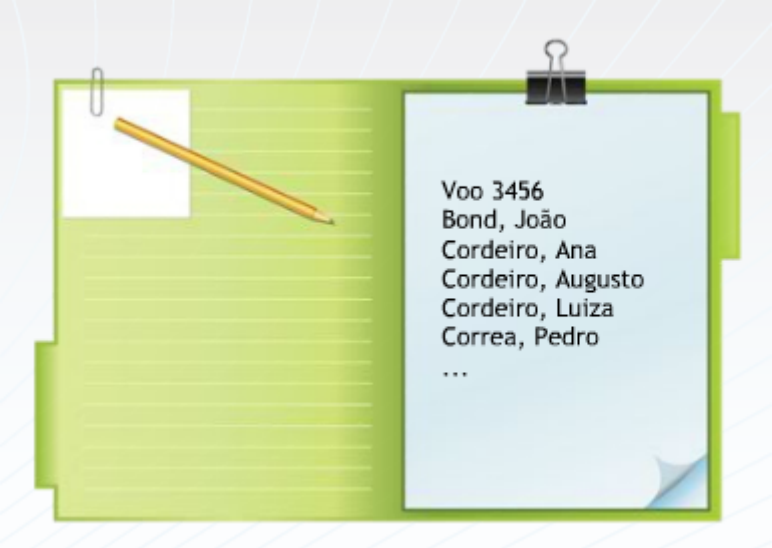
A ***inserção*** de um elemento na Fila acontece ***sempre ao final.***

A ***remoção*** de um elemento da Fila acontece ***sempre no início.***

Não é difícil de fazer uma analogia com estrutura da fila e as filas que enfrentamos no nosso dia-a-dia: a fila do banco, a fila para comprar um ingresso no cinema, etc.

Se pensarmos em termos de um Sistema Operacional para gerenciar a impressão de vários documentos, também fica fácil entendermos a filosofia da estrutura da fila: o primeiro documento a chegar é o primeiro a ser impresso se ignorarmos prioridade.

Sr. Augusto entregou as passagens e identidades de todos à atendente e ela foi pesquisar na lista de passageiros que era ordenada por sobrenome.



Pesquisar

No nosso curso, além de estudarmos as Estruturas de Dados, teremos outros tópicos necessários para que possamos implementar TDAs (TADs).

Estudaremos dois métodos de pesquisa: sequencial e binária. A pesquisa sequencial é indicada para listas que não estão ordenadas enquanto que a pesquisa binária, mais rápida, é perfeita para listas ordenadas.

Lista

A **lista** era ordenada por sobrenome senão a pesquisa seria muito demorada. Uma **lista** não precisa ser ordenada, depende da aplicação. Quando você vai ao mercado, você faz uma **lista** do que precisa comprar que não está em ordem alfabética. A **lista** é uma das Estruturas de Dados mais simples para agrupar dados, formando um conjunto com elementos do mesmo tipo e preservando a relação de ordem linear entre os elementos. Os elementos de uma **lista** são chamados de nós ou nodos e operações poderão ser feitas com as listas.

As formas de inserção e remoção dos elementos da lista é que definirão os tipos de listas (pilha, fila).

Tipos de Listas lineares

Sequencial - os dados são armazenados na Memória Principal de forma contigua. Como exemplo: as matrizes.

Encadeada - os dados vão sendo armazenados na memória em posições não adjacentes (não contíguas). Como é isso? Só quando estudarmos Alocação Dinâmica na Memória.

Ordenação

Classificar os dados é uma necessidade na maioria dos programas. Teremos a oportunidade de conhecer alguns métodos de ordenação e compará-los.

Depois que o *check in* foi feito, as malas foram para a esteira.

E foram sendo **empilhadas** à medida que iam chegando.

Pilha

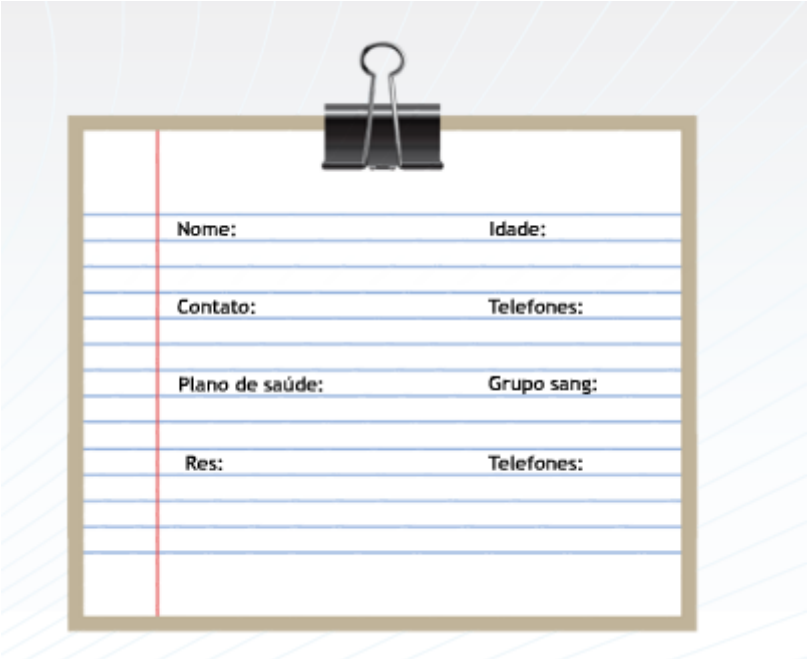
A pilha é um tipo de Estrutura de Dados onde a inserção de um elemento sempre acontece no topo da pilha e a remoção, também.

Pelo fato da inserção e remoção acontecerem na mesma extremidade da pilha, seu método de acesso é do tipo **LIFO** - *lost in, first out* (o último elemento que entra é o primeiro que sai).

Esta estrutura é muito usada pelo Sistema Operacional nas chamadas de funções onde o endereço de retomo é empilhado.

Passado algum tempo, os passageiros do voo 3456 foram chamados para embarcar e lá foi Sr. Augusto e sua família rumo ao Rio Ailã com parada em Fortaleza. A viagem nem tinha começado e nós, por aqui, já tínhamos começado a nos familiarizar com toda essa nomenclatura. No avião, os netos do Sr. Augusto ficaram encantados com tudo, visto que nunca tinham entrado em um avião. Depois que guardaram as malas de mão, ficando somente com as mochilas, começaram a conversar sobre a viagem e nesse momento, Sr. Augusto retirou da sua bolsa cinco fichas e foi distribuindo para cada um.

Nossa, vovô, você pensa em tudo! Falou um dos netos. E foram preencher a ficha com a ajuda das mães.



Struct

Podemos definir um **struct** como sendo um conjunto de elementos geralmente agrupados sob uma lógica e associados por um nome. Esses elementos podem ser variáveis simples, matrizes, outras estruturas e até funções.

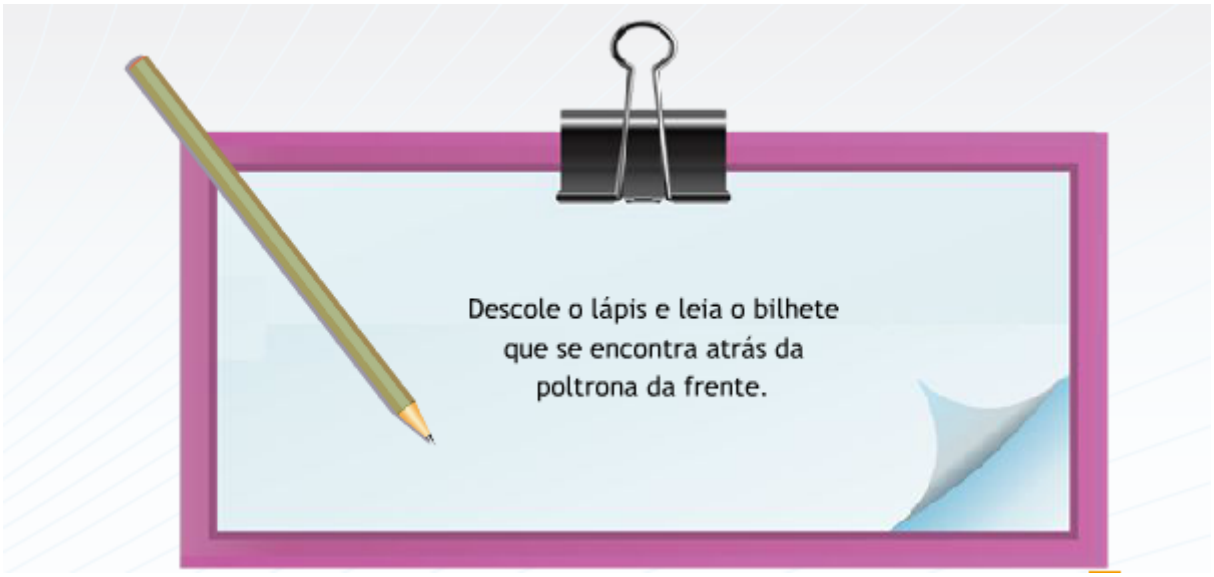
Por essa definição, podemos concluir que um **struct** pode ser formado por elementos de tipos diferentes. Cada elemento da estrutura é chamado de *membro ou campo.*

Observe a ficha que o Sr. Augusto deu para todos preencherem. Ela tem 8 membros dos quais 5 são do tipo vetor de char e três do tipo *int* (telefones e idade) ou então 7 do tipo vetor de char porque se colocarmos o ddd, o número ficará muito grande para ser armazenado em uma variável do tipo *int.*

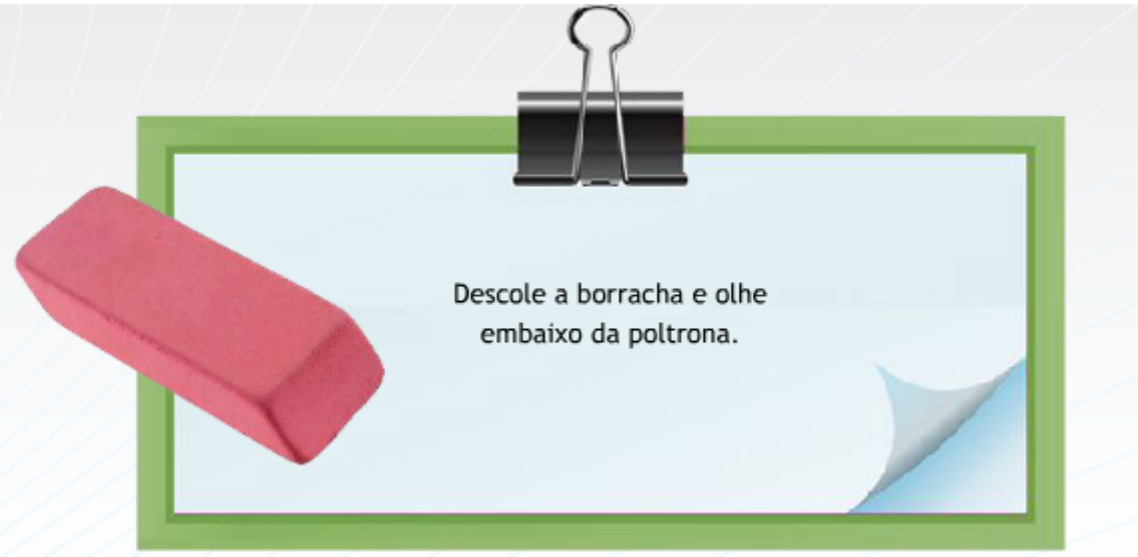
De qualquer maneira, estamos criando um novo tipo de "variável", visto que ela tem membros com tipos diferentes. Como acessar esses membros? Só lendo a aula de **struct,** é claro!

Quando terminaram de preencher a ficha, Sr. Augusto, que já gosta de uma brincadeira, falou para os netos: Abram a mochila que deixei algo para vocês lá.

Os netos já sabiam que alguma gracinha estava por vir. Abriram a mochila e encontraram a ficha abaixo:

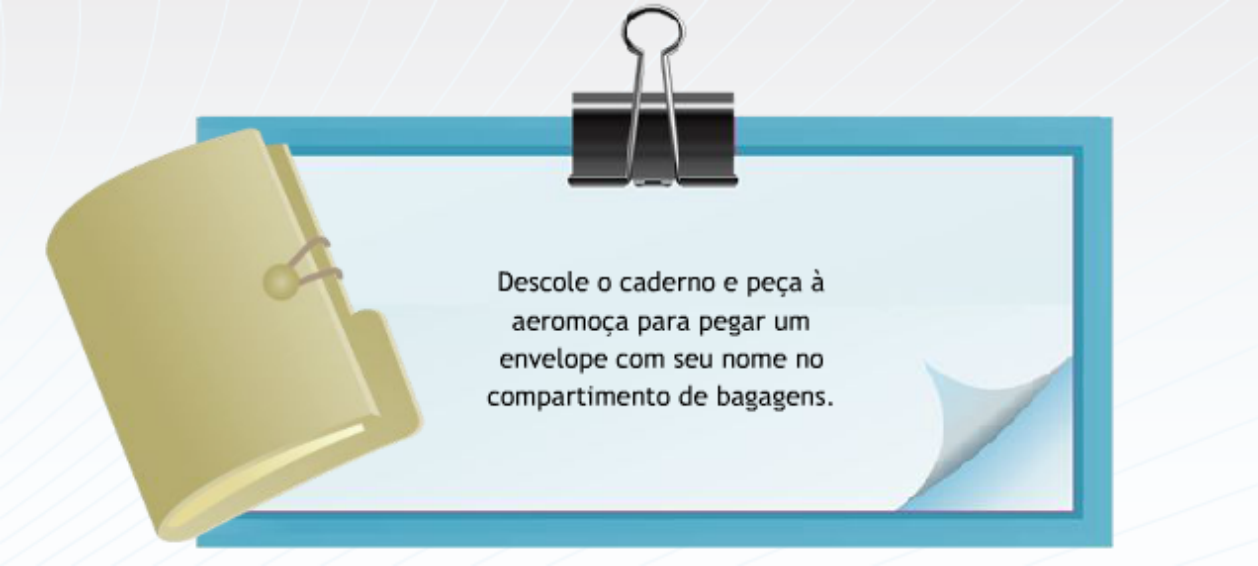


De um lado, algo para retirar, um lápis, e, do outro, um novo lugar para encontrar um outro bilhete.



Desta vez, tinha uma borracha e um outro lugar para encontrar um outro bilhete.

E lá foram eles, morrendo de curiosidade, buscar esse bilhete.



Desta vez, tinha um pequeno caderno e o lugar onde estava um envelope.

Chamaram a aeromoça e pediram, por favor, para que ela pegasse dois envelopes no compartimento de bagagens. Eles agradeceram e, rapidamente abriram os envelopes.

Dentro de cada envelope, estava escrito qual seria a função de cada um dos netos nessa viagem porque dividindo tarefas fica mais fácil para todos e o resultado é muito melhor disse o avô. Ele também falou que todo dia ele gostaria que eles, no café da manhã falassem se tinham cumprido suas tarefas no dia anterior. Abaixo, a função de um dos netos.

Descarregar a máquina todos os dias e fazer backup no pen drive.

Função

Na disciplina de Algoritmos, construímos programas só com uma *função*. Nesse período, aprenderemos a dividir o programa em partes menores com funções muito específicas.

Uma *função* é um bloco contendo cabeçalho, início e fim. Ela é identificada por um nome que deverá seguir as regras usadas para se nomear variáveis.

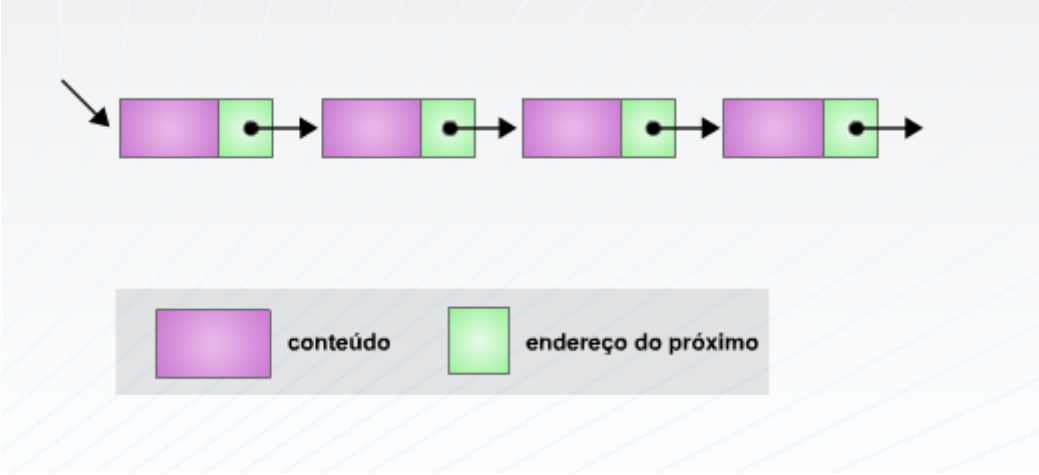
A função serve para executar tarefas menores como ler, calcular, determinar o maior/menor valor entre uma lista de valores, ordenar, converter para maiúsculas, entre outras. Após executar estas tarefas menores, a função retoma, ou não, um determinado valor para a função chamadora.

Observe um exemplo de função criada pelo usuário que exibe 50 asteriscos na tela toda vez que é chamada.

```
void asterisco()
{
 int x;
 for(x=1;x<=50;x++)
 cout<<"*";
}
```

Quer aprender a construir as suas funções, leia a próxima aula!

Eles acharam a brincadeira muito legal, pois era uma lista de tarefas toda **encadeada** porque cada vez que eles iam a um local (*endereço*) tinha algo a receber (*valor*) e uma dica para outro lugar (*um endereço*). Gostaram tanto da ideia que vão adaptar para brincar de detetive com os amigos.



A **lista encadeada**, também conhecida como lista ligada, se caracteriza por não ter seus nós, obrigatoriamente, alocados de forma contígua.

Como não existe garantia de uma alocação contigua, cada nó da lista leva o endereço do próximo nó para não perder o encadeamento.

O estudo de ponteiros é fundamental para listas encadeadas.

Representamos a lista por um ponteiro para o primeiro nó e, do primeiro ao último, segue-se o encadeamento através dos ponteiros presentes em cada nó.

O último elemento da lista aponta para NULL.

A viagem continuou calmamente até que chegaram ao aeroporto de Fortaleza. Desembarcaram, pegaram as malas e foram para o hotel. Chegando lá, ficaram espantados com o tamanho dos quartos e a vista para a praia e logo pediram para ir à feirinha fazer umas comprinhas. Coisa de turistas. Mas isso é uma outra história que não nos interessa no momento.

Sabemos que eles foram até a nascente do Rio Ailã no Morro do Caburaí em Roraima. Depois visitaram Brasília e finalmente chegaram ao Arroio Chuí no Rio Grande do Sul, atravessando o Brasil de Norte a Sul. Retornaram ao Rio de Janeiro por esses dias e perguntaram por vocês. Eles queriam saber se tinham aproveitado a viagem que fizeram com eles até Fortaleza e como andava o conhecimento sobre as Estruturas de Dados. Mandaram um recado: "Da próxima vez, aproveitem mais".

Nesta aula, iniciamos o estudo sobre as Estruturas de Dados. Temos consciência do nível de responsabilidade do ensino dessa disciplina, visto que não é suficiente apresentar as estruturas e as operações que podem ser realizadas com cada uma delas, mas ajudá-lo a desenvolver uma lógica de programação para que você se torne capaz de sair do abstrato e ir para o concreto, conhecendo e implementando ações para as estruturas que vamos explorar. Talvez seja uma das disciplinas mais difíceis do curso, mas tenha certeza de que estamos aqui para explicar seu conteúdo da forma mais simples possível nem que seja com uma história como foi a da aula de hoje. Sabemos que o conteúdo de Estrutura de Dados é muito amplo e não teríamos como ministrá-lo em uma disciplina, mas lhe daremos subsídios para que você possa se aprofundar sobre o assunto. Esta disciplina será um desafio, mas vamos vencê-lo!

O que vem na próxima aula

Na próxima aula, você estudará os seguintes assuntos:

- Conceito de funções;
- Funções;
- Funções com e sem retorno, com e sem passagem de parâmetros;
- Escopo de variáveis (global e local);

- Funções tendo vetores como parâmetros;
- Biblioteca de funções.

CONCLUSÃO

Nesta aula, você:

- Aprendeu as diferenças entre as Estruturas de Dados que estudaremos em nosso Curso tais como Fila, Lista, Pilha;
- Compreendeu o conceito de função que será de muita importância nesse curso;
- Compreendeu o conceito de struct;
- Compreendeu o conceito de Ordenação e de Pesquisa.