

Disciplina: Introdução a Programação

Aula 4: Comandos de entrada e saída

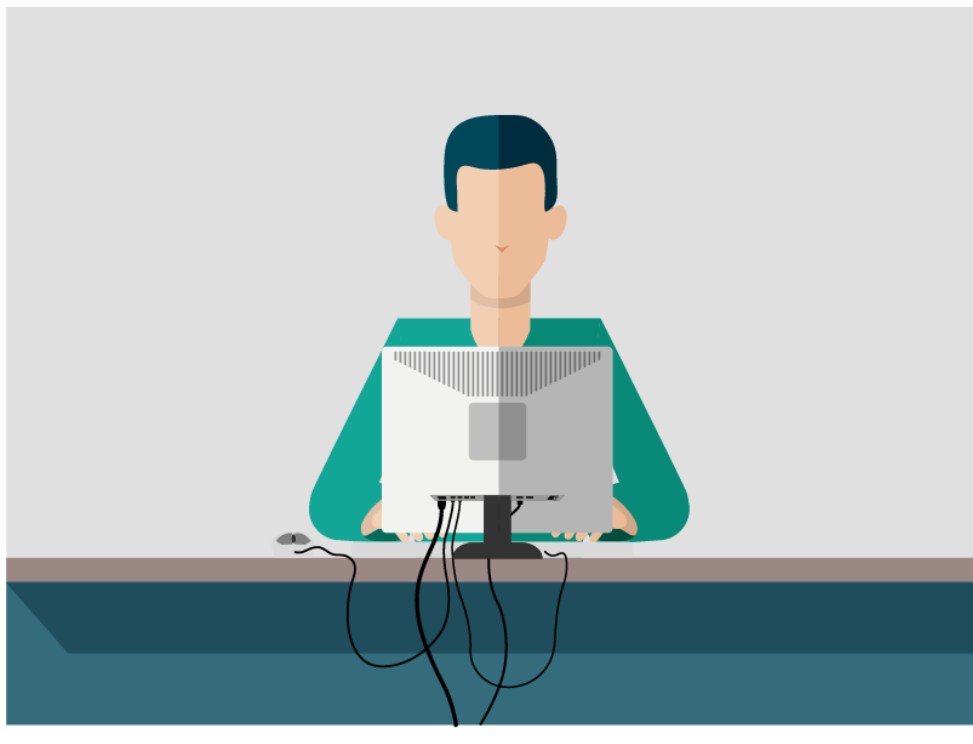
Apresentação

Nesta aula, estudaremos os primeiros comandos para a comunicação entre o usuário e o computador. São os comandos de entrada e saída que permitem que os usuários forneçam dados para o computador (comandos de entrada) e o computador exiba informações para os usuários (comandos de saída).

Os comandos de entrada e saída estarão presentes em todos os outros programas e algoritmos que escreveremos em nossas aulas. Por isso, você precisa dominá-los com segurança para que não sinta dificuldades mais adiante. Fique atento aos exemplos e realize todos os exercícios para conseguir compreender a forma como esses comandos são utilizados e a maneira como se comportam quando os programas são executados.

Objetivos

- Identificar os comandos de entrada e saída;
- Analisar problemas simples que possam ser resolvidos com programas;
- Aplicar algoritmos e programas para solução dos problemas apresentados.



Comandos de entrada e saída

Uma característica interessante da programação é que ela permite a interação entre a máquina – no nosso caso, o computador – e seus usuários.

Essa interação ocorre por meio dos comandos de entrada e saída (ou comandos de I/O – diz-se “ai/ou”, pois o termo vem das palavras em inglês input e output).

Nesta disciplina, trabalharemos apenas com os comandos de entrada de dados via teclado, mas um input pode vir de outras origens, como o clique em um botão do mouse ou um sinal detectado por um sensor de movimentos, por exemplo.

Os comandos de saída podem direcionar a informação para uma impressora ou um dispositivo de áudio, por exemplo, mas o destino padrão das saídas de nossos programas e algoritmos será o monitor de vídeo.

Dica

Conforme já estudamos, nosso programa para teste dos algoritmos que construiremos é o [Portugol Studio](http://lite.acad.univali.br/portugol).
<\\lite.acad.univali.br/portugol>

Você já fez o download do programa e o instalou no seu computador? Se já fez, ótimo! Se não, faça isso o quanto antes, pois você vai começar a utilizar o Portugol Studio muito em breve.

E o ambiente integrado de desenvolvimento, o [o DevC++](https://sourceforge.net/projects/orwelldevcpp/files/latest/download)
<https://sourceforge.net/projects/orwelldevcpp/files/latest/download>, que vamos utilizar para testar os programas? Você já fez o download dele?



Atenção

Antes de iniciarmos a prática, é preciso que você tenha em mente que começaremos escrevendo soluções bastante simples, que se tornarão mais complexas à medida que novos comandos e estruturas forem aprendidos. Construa uma base sólida de conhecimentos para programação.

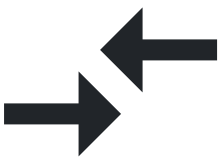
Comandos de entrada e saída em Portugol

Geralmente, os algoritmos utilizam os comandos **leia** e **escreva** para **entrada** e **saída de dados**, respectivamente.

Algumas vezes, vemos pequenas variações dessas instruções:

COMANDO DE ENTRADA

pode aparecer como **receba**, **receber** ou **ler**.



COMANDO DE SAÍDA

Comando de saída pode passar a ser **exiba**, **exibir** ou **escrever**.

Como seguiremos a sintaxe do Portugol Studio, utilizaremos **leia** e **escreva**, sem nenhuma variação.

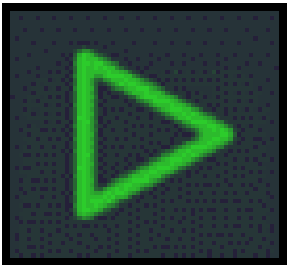
Dica

A sintaxe de uma linguagem de programação define a forma como seus comandos e estruturas devem ser utilizados.

Para termos uma ideia de como são utilizados esses comandos, imagine que você deseja criar um algoritmo que, a partir do ano atual e do ano de nascimento, informe a idade aproximada de uma pessoa. Veja uma solução possível, escrita no Portugol Studio:

```
programa
{
funcao inicio()
{
inteiro anoNasc, anoAtual, idade
escreva("Ano atual: ")
leia(anoAtual)
escreva("Ano de nascimento: ")
leia(anoNasc)
idade=anoAtual-anoNasc
escreva("Você tem ou fará ", idade, " anos")
}
}
```

Quando terminar de digitar seu algoritmo, execute-o pressionando as teclas Shift+F6 ou clicando no botão reproduzido a seguir:

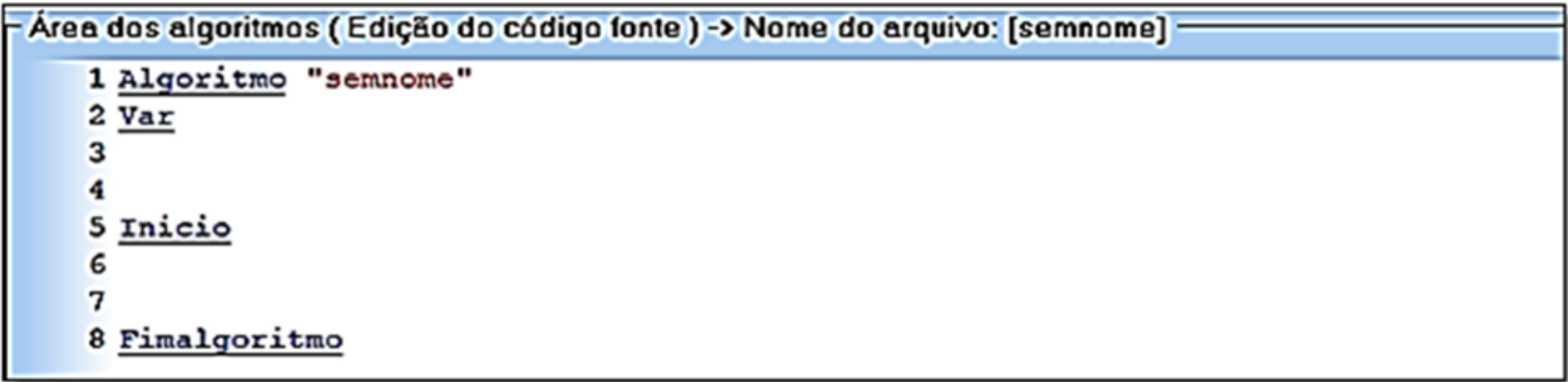


Botão para executar o algoritmo.

Quando você criar um novo programa no Portugol Studio, as linhas 1, 2, 3, 4, 12 e 13 serão escritas automaticamente.

Elas marcam as estruturas inicial (linhas 1, 2, 3 e 4) e final (linhas 12 e 13) do algoritmo e você não precisa se preocupar com elas, pois são uma particularidade dos Portugol Studio.

O **VisulAlg**, que é uma outra ferramenta para execução de algoritmos, tem outra estrutura:



Vamos ver o que fazem as outras linhas:

Linha 5	inteiro anoNasc, anoAtual, idade	Cria as variáveis anoNasc, anoAtual e idade e as declara como sendo tipo inteiro. Essas variáveis servirão para o armazenamento do ano de nascimento do usuário, do ano atual e da idade (que será calculada pelo algoritmo).
Linha 6	escreva("Ano atual: ")	O comando escreva é o responsável por exibir em tela mensagens e conteúdo de variáveis e constantes. Neste exemplo, ele exibe a mensagem entre aspas na tela.
Linha 7	leia(anoAtual)	O comando leia é o responsável por receber valores informados pelo usuário via teclado e armazená-los em variáveis. Aqui, o valor numérico informado pelo usuário será armazenado na variável anoAtual.
Linha 8	escreva("Ano de nascimento: ")	Esta linha exibirá em tela a mensagem entre aspas.
Linha 9	leia(anoNasc)	Receberá o valor numérico informado pelo usuário e armazenará na variável anoNasc.
Linha 10	idade=anoAtual-anoNasc	Esta expressão fará a subtração definida à direita do sinal de atribuição (“=”) e armazenará o resultado da mesma na variável idade.
Linha 11	escreva("Você tem ou fará ", idade, " anos")	Esta linha exibirá em tela a mensagem “Você tem ou fará XX anos”, em que XX será o valor armazenado na variável idade. Veja que combinamos, em um mesmo comando escreva, a exibição de mensagens e do conteúdo de uma variável. Para isso, usamos as aspas duplas para delimitar as mensagens e usamos a vírgula para separar mensagens de variáveis.

Imagine que você viajou para os Estados Unidos e está muito confuso com o modo que os americanos usam para medir a temperatura, em Fahrenheit. Para facilitar, você decide criar um algoritmo que, dada uma temperatura em Fahrenheit, exiba seu equivalente em graus Celsius. Uma solução possível seria:

```
programa
{
funcao inicio()
{
real tempF, tempC
escreva("Temperatura em Fahrenheit: ")
leia(tempF)
tempC=(tempF-32)/1.8
escreva("Temperatura em graus Celsius: ", tempC)
}
}
```

Vamos ver o que fazem as linhas de código?

Linha 5 `real tempF, tempC`

Cria as variáveis `tempF` e `tempC`, e as declara como sendo do tipo `real`. Essas variáveis irão armazenar, respectivamente, a temperatura em graus Fahrenheit e a temperatura em graus Celsius.

Linha 6 `escreva("Temperatura em Fahrenheit: ")`

Esta linha faz com que seja exibida em tela a mensagem entre aspas.

Linha 7 `leia(tempF)`

Permite que o usuário informe um valor via teclado e faz com que esse valor seja armazenado na variável `tempF`.

Linha 8 `tempC=(tempF-32)/1.8`

Esta expressão fará a conversão do valor em graus Fahrenheit para graus Celsius. O resultado é armazenado na variável `tempC`.

Linha 9 `escreva("Temperatura em graus Celsius: ", tempC)`

Esta linha é responsável por mostrar, em tela, a mensagem entre aspas seguida do valor armazenado na variável `tempC`. Combinamos, em um mesmo comando `escreva`, a exibição de mensagens e do conteúdo de uma variável. Perceba o uso da vírgula separando os itens.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Arredondando valores reais no Portugol Studio

Ao testar o algoritmo de conversão de temperaturas, é possível que você tenha obtido um valor com várias casas decimais como resultado.

Nem sempre é uma boa opção informar ao usuário esse tipo de valor. O ideal é que o número seja arredondado para mostrar um número menor de casas decimais – uma ou duas – e, assim, facilitar sua compreensão.

Para arredondarmos valores no Portugol Studio e configurar o número de casas decimais que desejamos exibir, é preciso utilizar a função `arredondar`, da biblioteca `Matematica-->mat`. Veja como fazer:

```
real pi_Longo = 3.14159265358979323846
real pi_Curto
// Arredonda o valor de pi_Longo para 2 casas decimais
pi_Curto = mat.arredondar(pi_Longo,2)
```

Para que seja utilizada a função **arredondar**, precisamos incluir a biblioteca **Matematica-->mat**. Veja como fazer:

```
programa
{
inclua biblioteca Matematica --> mat
funcao inicio()
{
...
}
}
```

Se você quiser utilizar a formatação de casas decimais no problema de conversão de temperaturas, insira a **linha de inclusão da biblioteca no início do algoritmo** e altere a linha de exibição da temperatura em Celsius para que fique assim:

```
escreva("Temperatura em graus Celsius: ", mat.arredondar(tempC,2))
```

Você percebeu, no exemplo da função **arredondar**, que uma linha do algoritmo começa com os caracteres `“//”`? Trata-se de uma linha de comentário.

Linhas de comentário são muito úteis, pois nos permitem escrever informações que nos ajudam a compreender melhor o código. Elas não serão executadas pelo computador e servem somente para fins de documentação.

Comandos de entrada e saída em C++

Na linguagem de programação C++, os comandos cin e cout são os que utilizaremos para entrada e saída de dados, respectivamente. Antes de falarmos mais sobre eles, é preciso saber que os programas que escreveremos em C++ terão uma estrutura básica que estará presente em todas as soluções:

```
#include < iostream >
using namespace std;
int main()
{
$
}
```

O código será escrito entre as chaves, que marcam o início e o fim do bloco de comandos pertencentes ao seu programa (a partir da linha 5). Na linha 1, temos a diretiva include, que informa quais bibliotecas de comandos serão necessárias no programa. Para o exemplo anterior, somente utilizaremos a biblioteca **iostream**, que contém os recursos básicos para entrada e saída e nos permitirá utilizar os comandos **cin** e **cout**.

Você se lembra do algoritmo que fizemos no início desta aula, para cálculo da idade do usuário a partir do ano atual e do ano de nascimento? Vamos ver como ele ficaria escrito em C++?

```
#include < iostream >
using namespace std;
int main()
{
int anoAtual, anoNasc, idade;
cout <<"Ano atual: ";
cin >> anoAtual;
cout <<"Ano de nascimento:";
cin >> anoNasc;
idade=anoAtual-anoNasc;
cout <<"Você tem ou fará " << idade << " anos.";
}
```

Atenção

Quando terminar de digitar seu programa, compile-o e execute-o pressionando a tecla F11.

Dica

C++ é uma linguagem de programação compilada. O compilador é responsável por verificar erros de sintaxe no programa e, quando o programa está sintaticamente correto, produzir outro programa em linguagem de máquina.

Vamos ver o que fazem as linhas do programa?

Linha 1 #include <iostream>	Sinaliza a inclusão da biblioteca iostream , que permite o uso dos comandos cin e cout , entre outros.
Linha 2 using namespace std;	Um namespace é uma região declarativa que fornece um escopo para os identificadores (nomes dos tipos, função, variáveis etc.) dentro dele. Todas as declarações dentro desses blocos são declaradas no escopo nomeado. Utilizar esta linha no início do programa faz com que não precisemos escrever std::cout <<"Ano atual: " .
Linha 3 int main()	Os programas em C são organizados em forma de função. A função principal é denominada main , esta linha marca a abertura da função principal do programa.
Linha 4 {	Marca o início do programa.
Linha 5 int anoAtual, anoNasc, idade;	Declara as variáveis anoAtual , anoNasc e idade , e as declara como sendo do tipo inteiro. Essas variáveis irão armazenar, respectivamente, o ano atual, o ano de nascimento do usuário e a idade a ser calculada pelo programa.
Linha 6 cout <<"Ano atual: ";	Exibe, em tela, a mensagem entre aspas.
Linha 7 cin >> anoAtual;	Recebe, via teclado, um valor e o armazena na variável anoAtual .
Linha 8 cout <<"Ano de nascimento:";	Exibe, em tela, a mensagem entre aspas.
Linha 9 cin >> anoNasc;	Recebe, via teclado, um valor e o armazena na variável anoNasc .
Linha 10 idade=anoAtual-anoNasc;	Calcula a idade do usuário realizando a subtração entre o ano atual e ano de nascimento e armazena na variável idade .
Linha 11 cout <<"Você tem ou fará " << idade << " anos.";	Exibe, em tela, as mensagens entre aspas combinadas com o valor armazenado na variável idade .
Linha 12 }	Marca o fim do programa.

Vamos ver como ficaria, em C++, o algoritmo para conversão de temperaturas?

```
#include < iostream >
using namespace std;
int main()
{
float tempF, tempC;
cout << "Temperatura em Fahrenheit: ";
cin >> tempF;
tempC=(tempF-32)/1.8;
cout << "Temperatura em graus Celsius: " << tempC;
}
```

Veja o que fazem as linhas de código do programa anterior:

Linha 5	float tempF, tempC	Cria as variáveis tempF e tempC , e as declara como sendo do tipo real. Essas variáveis irão armazenar, respectivamente, a temperatura em graus Fahrenheit e a temperatura em graus Celsius.
Linha 6	cout << "Temperatura em Fahrenheit: ";	Esta linha faz com que seja exibida em tela a mensagem entre aspas.
Linha 7	cin >>tempF;	Permite que o usuário informe um valor via teclado e faz com que esse valor seja armazenado na variável tempF .
Linha 8	tempC=(tempF-32)/1.8;	Esta expressão fará a conversão do valor em graus Fahrenheit para graus Celsius. O resultado é armazenado na variável tempC .
Linha 9	cout >> "Temperatura em graus Celsius: " >> tempC;	Esta linha é responsável por mostrar, em tela, a mensagem entre aspas seguida do valor armazenado na variável tempC . Combinamos, em um mesmo comando escreva, a exibição de mensagens e do conteúdo de uma variável. Perceba o uso da vírgula separando os itens.

Exibindo a acentuação gráfica no C++

- Se você usou acentuação gráfica nos programas que escreveu em C++, percebeu que, quando executado, o programa substituiu os caracteres acentuados por alguns símbolos.
- Para que isso não aconteça, é preciso que você inclua a biblioteca locale e configure o idioma para português do Brasil.

Observe como ficaria o início de um programa que realiza essa configuração:

```
#include < iostream >
#include < locale >
using namespace std;
int main()
{
    setlocale(LC_ALL, "ptb");
    ...
}
```

Dica

C++ é case sensitive, portanto, escreva as letras maiúsculas e minúsculas exatamente como vê no exemplo.

Agora que você já conhece os comandos de entrada e saída, vamos praticar? Você está construindo o alicerce de seu aprendizado sobre programação de computadores.

Atividade

- 1) Leia os enunciados a seguir e crie os algoritmos que solucionam os problemas propostos.
 - a) Exiba o quadrado e o cubo de um número inteiro qualquer informado pelo usuário.
 - b) Receba a quantidade de dias que uma pessoa já viveu e informe quantos anos ela tem. Desconsidere os anos bissextos.
 - c) Sabendo que o dono de uma loja aplica 20% sobre o valor de compra de um produto, informe o valor sugerido para venda com base no valor pago na compra do produto.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Atividade

2) Com base nos algoritmos escritos na atividade anterior, escreva os programas em C++. Abaixo estão os enunciados.

a) Exiba o quadrado e o cubo de um número inteiro qualquer informado pelo usuário.

b) Receba a quantidade de dias que uma pessoa já viveu e informe quantos anos ela tem. Desconsidere os anos bissextos.

c) Sabendo que o dono de uma loja aplica 20% sobre o valor de compra de um produto, informe o valor sugerido para venda com base no valor de compra do produto.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Observe o algoritmo a seguir:

```

programa
{
inclua biblioteca Matematica --> mat
funcao inicio()
{
real qtdeDin, valComb, qtdeLitros
escreva("Quanto custa o litro do combustível? R$ ")
leia(valComb)
escreva("Quantos reais você tem? R$ ")
leia(qtdeDin)
qtdeLitros=qtdeDin/valComb
escreva("Com R$ ", qtdeDin, " você consegue abastecer ",
mat.arredondar(qtdeLitros,2)," litros de combustível.")
}
}

```

Sabendo que o valor do litro do combustível é R\$ 3,90 e que você dispõe de R\$ 200,00 para abastecer, qual mensagem seria exibida pelo computador na linha 12?

Notas

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma
 simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria
 tipográfica e de impressos.

Título modal¹

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

Referências

MANZANO, J. A. N. G., OLIVEIRA, J. F. Algoritmos: lógica para desenvolvimento de programação de computadores. 28.ed. São Paulo: Érica, 2016.

PUGA, S.; RISSETTI, G. . São Paulo: Prentice Hall, 2005.Lógica de programação e estruturas de dados com aplicações em Java. 2.ed. São Paulo: Prentice Hall, 2005.

Próxima aula

- Estruturas seletivas;
- Algoritmos e programas utilizando as estruturas de seleção;
- Resultados produzidos por códigos que empregam estruturas seletivas.

Explore mais

- Assista ao [video <https://www.youtube.com/watch?v=JfJU2ZC8wV4 >](https://www.youtube.com/watch?v=JfJU2ZC8wV4) sobre os comandos de entrada e saída no Portugol Studio. A versão do software utilizada no vídeo é mais antiga do que a versão atual, mas os comandos e sua utilização permanecem os mesmos