

# Programação Básica

---

Estrutura de um algoritmo

# Código-fonte

---

- Como vimos na aula anterior um algoritmo pode ser representado usando um fluxograma
- Um algoritmo pode também ser representado usando texto
- Esse texto porém deve ser apresentado de uma forma clara e estruturada
- Para que possa ser traduzido para a linguagem de máquina (binário)

# Variáveis

---

- São áreas da memória do computador para armazenar valores
- No início do seu programa, defina uma variável para cada valor que você precisar usar.
- Exemplo:
  - Para calcular a idade de uma pessoa precisamos do ano que ela nasceu
    - `ano` será uma variável do nosso algoritmo
  - Para calcular a média semestral quais serão as nossas variáveis?

# Variáveis

---

- Para que o computador saiba quanto espaço precisa para armazenar as informações da variável precisamos dizer o tipo de dado
- Diferente das variáveis da matemática (que só guardam números) aqui podemos guardar números **inteiros**, **números reais**, **texto** e **valores lógicos**
- Então sempre que declararmos uma variável devemos dizer o seu tipo.

# Variáveis

---

- Tipos de variáveis:
  - inteiro – números inteiros
  - real – números com casas decimais
  - caractere – textos
    - Os textos devem ser sempre cercados por aspas como no exemplo abaixo:
      - “Tadeu Ferreira Oliveira”
  - logico – pode ser VERDADEIRO ou FALSO

# Variáveis

---

- Identifique os tipos das seguintes variáveis:
  - altura
  - peso
  - idade
  - primeiroNome
  - nomeCompleto
  - estaChovendo
  - totalDePatos

# Nomes das variáveis

---

- Há algumas regras a seguir para o nome das variáveis:
  - Devem começar com letras minúsculas
  - Não podem conter espaços
  - Não podem conter acentos nem caracteres especiais (@#\$%^&\*)
  - Há ainda um conjunto de nomes chamados palavras reservadas, essas palavras tem significado especial para a linguagem então não podem ser usadas como nomes de variáveis

# Blocos de comandos

---

- Depois de decididas as variáveis, podemos inserir os comandos
- Todos os comandos devem estar dentro de um bloco de comandos.
- Na linguagem C os blocos de comandos são delimitados por
  - {
  - //Conjunto de comandos
  - }



# Leitura e Escrita

---

- O computador pode realizar operações de interação com o usuário a partir de 2 ações:
  - Leitura – Permite que o usuário digite um valor que será armazenado em uma variável
  - Escrita – Exibe na tela o valor de uma variável
- Para ler uma variável use:
  - **`scanf("expressão de controle",&variavel);`**
- Para escrever na tela o valor de uma variável use:
  - **`printf("expressão de controle",variavel);`**

# scanf

- O scanf permite a leitura de dados formatados
- Para isso na expressão de controle é necessário informar o tipo de dado a ser lido

Tipo	Dado	Tipo de variável a ser usado
%i	Um número inteiro	int
%f	Um número real (ponto flutuante)	float ou double
%c	Um único Caracter	char
%s	Uma cadeia de caracteres (string)	char [tamanho_maximo]

# printf

- Permite a saída de texto na tela ou para um arquivo
- Pode-se intercalar o texto com os valores de variáveis, para isso usaremos os marcadores semelhantes ao scanf

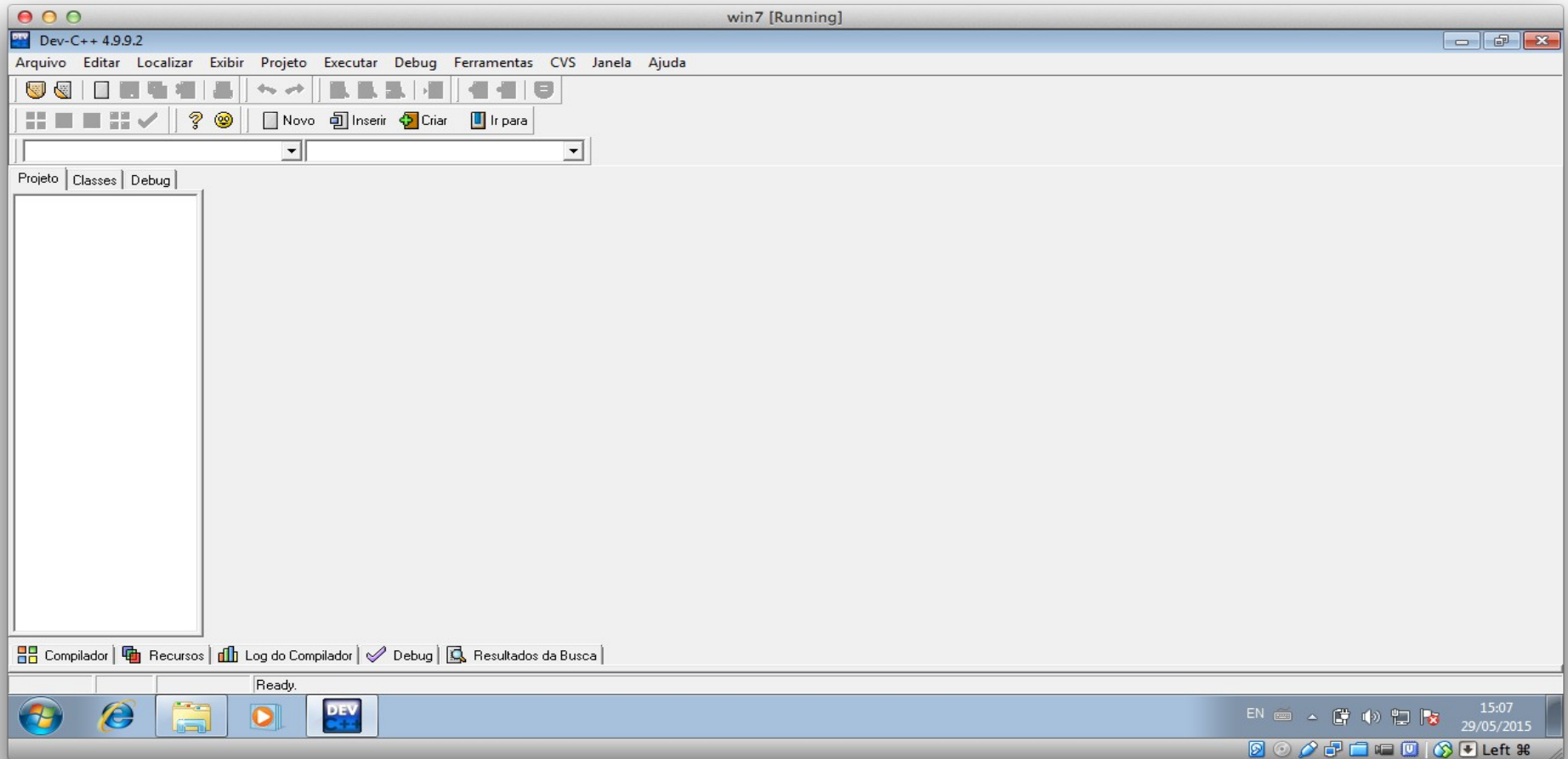
Tipo	Dado	Tipo de variável a ser usado
%i	Um número inteiro	int
%f	Um número real (ponto flutuante)	float ou double
%c	Um único Caracter	char
%s	Uma cadeia de caracteres (string)	char [tamanho_maximo]

# DevC++

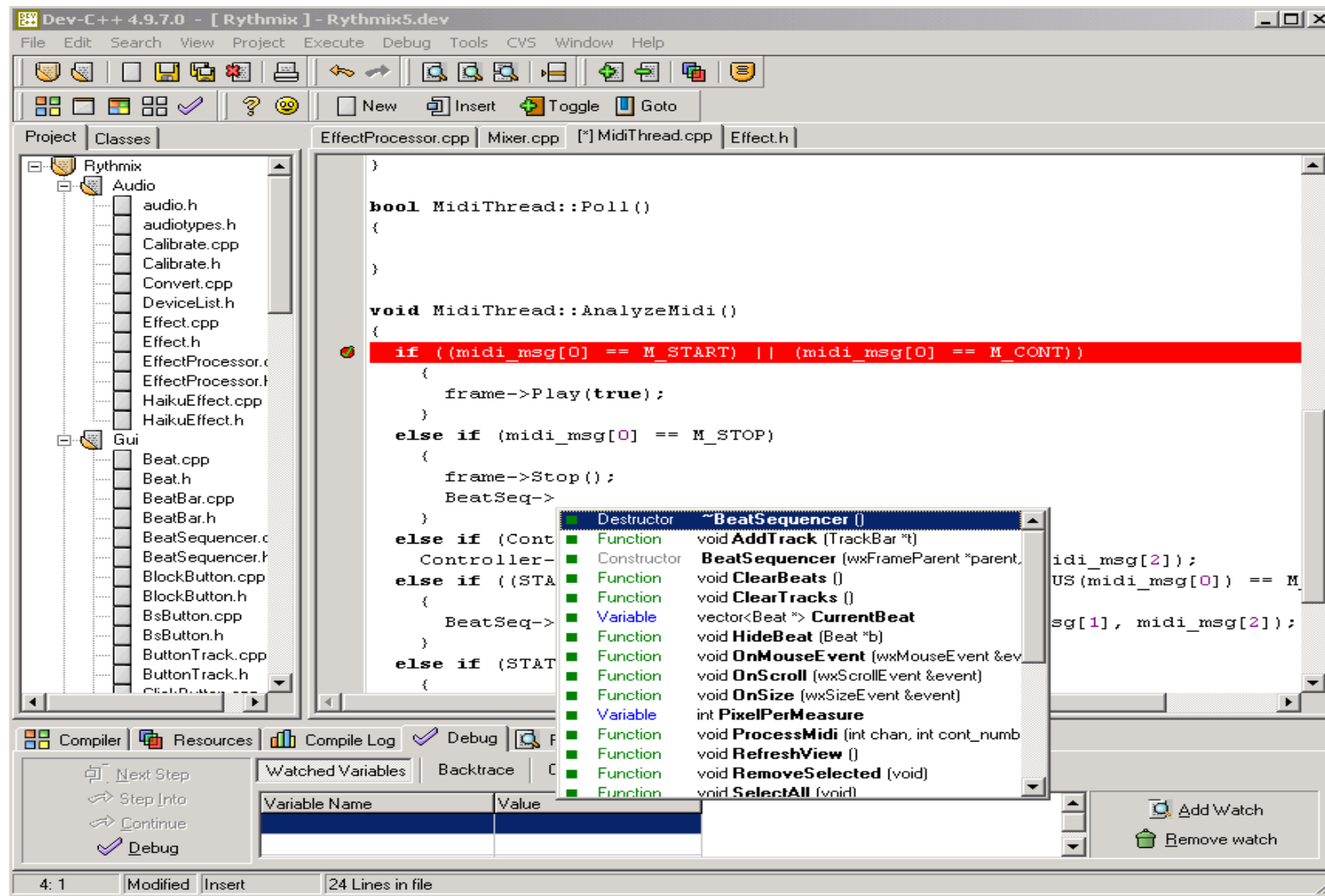
---

- A ferramenta que usaremos para escrever e testar nossos algoritmos é o DevC++
- Esta ferramenta permite escrever código em C ou C++
- Por trás desta interface está o compilador GCC
- Você pode usar qualquer editor de texto e o compilador em linha de comando diretamente
- Vejamos a interface do DevC++

# DevC++



# Barras de Ferramentas



# Barra de Ferramentas

The image shows the Dev-C++ 4.9.7.0 toolbar with four callout boxes pointing to specific icons:

- Compilar**: Points to the icon of a document with a lightning bolt.
- Executar**: Points to the icon of a document with a play button.
- Compilar e Executar**: Points to the icon of a document with a lightning bolt and a play button.
- Recompilar tudo**: Points to the icon of a document with a lightning bolt and a refresh symbol.

The toolbar also includes other icons for file operations (New, Open, Save, Print, Undo, Redo), editing (Copy, Paste, Find, Replace), and development (Toggle, Goto, New, Insert). The main window shows a project named 'Rythmix' with a file tree containing 'Audio', 'audio.h', and 'audiotypes.h'. The code editor displays the following code:

```
}  
  
bool MidiThread::Poll()  
{
```

# Hello World!

---

- Nosso Primeiro programa será um programa clássico conhecido por todos os programadores. O famoso
  - Hello World!
- O objetivo desse programa é escrever na tela a frase acima.
- Inicie o DevC++ e clique em novo arquivo



# Hello World!

---

- O seu código fonte será:

```
#include <stdio.h>
int main() {
    printf("Hello, World!");
}
```

# Hello World!

---

- Execute o seu programa pressionando F9

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    printf( "Hello, World!" );
    system( "pause" );
}
```

# Operações

---

- Assim como na matemática as variáveis podem ser usadas para efetuar operações
- Exemplo:
  - Como saber quantos anos terei no próximo ano?
  - idade + 1

# Operadores Aritméticos

---

Operador	Descrição	Exemplo
+	Soma	resultado = 5 + 5
-	Subtração	resultado = 5 - 5
*	Multiplicação	resultado = 5 * 5
/	Divisão	resultado = 5 / 5

# Exemplo

---

Faça um programa que leia o total de compras, o valor pago e exiba o troco que deve ser dado

# Atividade

---

1. Faça um programa que imprime o desenho a seguir:

```
  *  
 * * *  
* * * * *  
* * * * * * *
```

2. Faça um programa que leia a idade atual e escreva a idade do usuário no próximo ano

3. Faça um programa que leia um número e exiba o dobro desse número