

ESQUEMA AULA PRÁTICA 1

- ❑ Familiarização com o Ambiente de Desenvolvimento NetBeans
- ❑ Construção do primeiro programa em java.
- ❑ Introdução à Linguagem de Programação JAVA
 - 0 – Iniciar o ambiente de desenvolvimento integrado (Integrated Development Environment – IDE) NetBeans

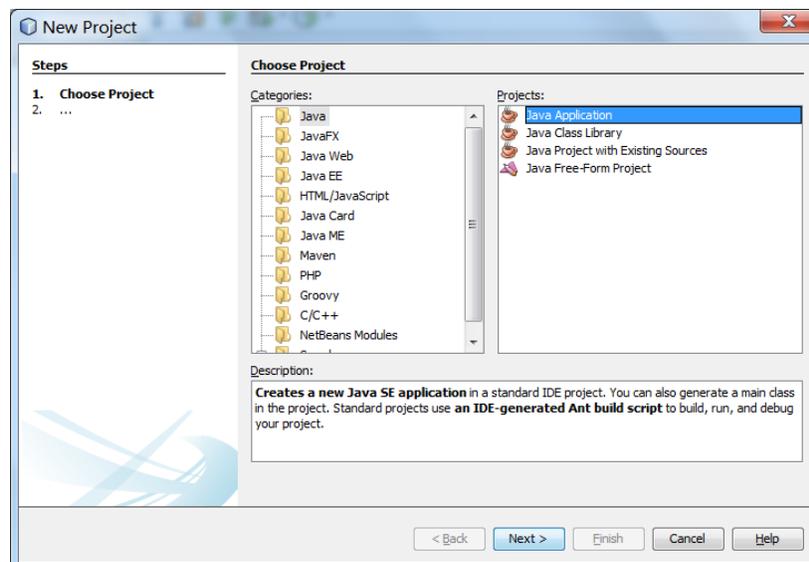
a) - Comece por criar uma diretoria com o nome POO onde irá colocar os trabalhos desta disciplina.

Se está a trabalhar no computador da sala de aulas no final da aula deverá copiar o seu trabalho para um suporte pessoal (ex.: PEN, conta na dropBox, conta no servidor da UBI, etc, ...)

b) - Selecione, Start | All Programs | Shortcut to NetBeans

- Inicie o NetBeans e no Menu File selecione a opção New Project.
Dependendo da versão do seu editor deverá ter uma das duas hipóteses seguintes:

Em Oracle NetBeans 8.2 (versão nas máquinas da sala de aula)

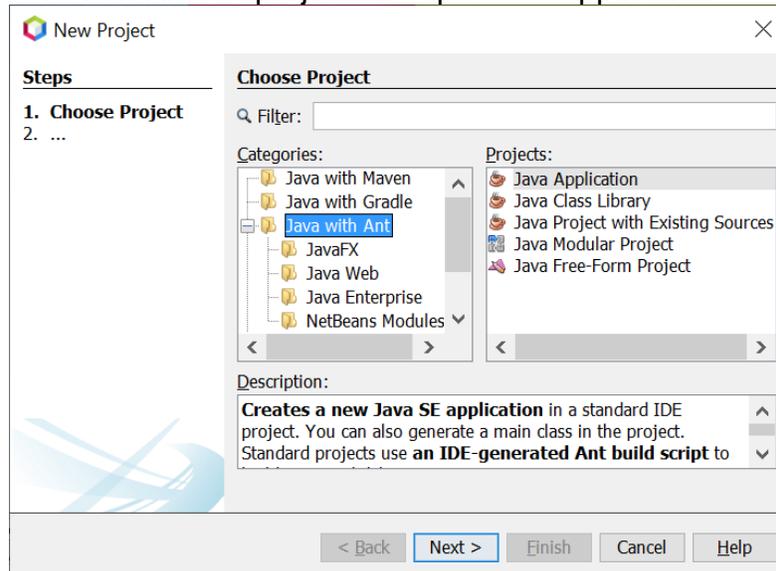


H1 - Selecionar a categoria Java e o projeto do tipo Java Application.

Ou

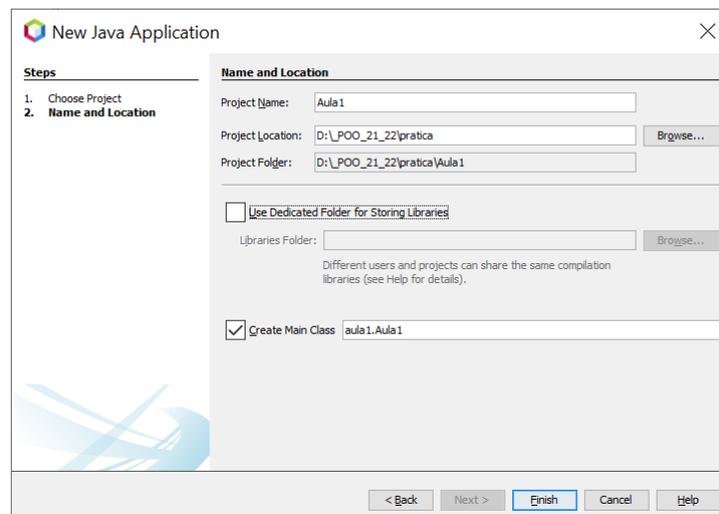
(Se no seu computador pessoal instalou o Apache NetBeans, versão 12.0)

H2 – Selecione Java with Ant e o projecto do tipo Java Application



c) Chame ao projeto Aula1 e em Project Location coloque a diretoria onde quer guardar o trabalho da disciplina de POO.

d) Se seleccionar “Create main Class” será criada uma classe com o mesmo nome que o projeto e com o cabeçalho (ou assinatura) do método main.



- Um programa em JAVA é uma classe.
- Cada aplicação terá que possuir uma classe onde está definido o método (ou função) **main** pelo qual se inicia a execução do programa.
- Por convenção o nome de uma classe começa por uma letra maiúscula.
- O **;** é um terminador de instruções.
- As chavetas, **{ }** são o delimitador de conjuntos de instruções.

Vai construir o seu primeiro programa Java, na classe Aula1 acabada de criar.

1 – Na classe Aula1 insira a instrução a bold do código abaixo.

```
public class Aula1 {  
    public static void main(String[] args) {  
        System.out.println("Este é o 1º programa em Java");  
    }  
}
```

- Para executar o programa, seleccionar a opção **run** no projeto Aula1. Pode seleccionar esta opção abrindo o menu de opções com o botão direito do rato, sobre o nome do ficheiro que contém o código ou directamente sobre o código.

Poderá observar o output na consola do editor.

Notas:

O ficheiro que contém a classe JAVA é gravado com a extensão .java; A compilação deste ficheiro produz um outro de extensão .class que poderá ser interpretado pelo interpretador de JAVA da sua máquina.

Uma ideia base da linguagem JAVA é a de que “um programa em JAVA deve poder ser executado em qualquer parte”. Assim o código fonte da aplicação é compilado para uma representação intermédia, independente do sistema de execução e da arquitectura da máquina. Essa representação intermédia é designada por byte-code.

De seguida este código pode ser interpretado sobre o ambiente de cada máquina específica. Para cada plataforma em que se pretende executar um programa em Java é necessário um “motor de execução” designado por Java Virtual Machine” (JVM). A JVM recebe byte-

code e transforma-o em instruções executáveis na máquina onde o ambiente Java é instalado.

O JAVA pode ser usado para criar dois tipos de programas: Aplicações e Applets.

Applets são porções de código Java não executável por si próprio. Requerem a existência de um “browser” que incorpore e execute a JVM.

2 – No mesmo projecto podemos ter outros programas, cada um com o seu método main.

Se isso acontecer, para executarmos um programa teremos de seleccionar a classe que contém o método main pretendido. A execução do programa terá início na primeira instrução do método main que terá de ter o cabeçalho:

public static void main (String[] args).

a) Crie uma nova classe com um nome à sua escolha, e nessa classe construa um programa que escreva no ecrã o texto abaixo que contém uma linha em branco entre as linhas de texto:

1º aula de POO

Estudo da instrução de escrita

b) Qual a diferença entre a instrução **System.out.println** e a instrução **System.out.print**?

3 – A linguagem Java é uma extensão da linguagem C. Além dos Objetos que iremos estudar nesta disciplina contém os tipos de dados que existem no C e que são designados por tipos primitivos da linguagem:

A tabela que se segue, contém para cada tipo primitivo, uma descrição dos seus valores, o valor por omissão, o número de bits em que são representados e o intervalo dos seus valores.

Tipos de Dados Primitivos

| Tipo | Valores | Valor por omissão | Nº de bits | Gama de valores |
|---------|-------------------|-------------------|------------|---------------------------|
| boolean | true, false | False | 1 | -- |
| char | carateres unicode | \u0000 | 16 | \u0000 a \uFFFF |
| byte | inteiro com sinal | 0 | 8 | -128 a +127 |
| short | inteiro com sinal | 0 | 16 | -32768 a + 32767 |
| int | inteiro com sinal | 0 | 32 | -2147483648 a +2147483647 |
| long | inteiro com sinal | 0 | 64 | -1E+20 a +1E+20 |
| float | IEEE 754 FP | 0.0 | 32 | ±3.4E+38 a ±1.4E-45 |
| double | IEEE 754 FP | 0.0 | 64 | ±1.8E+308 a ±5E-324 |

A declaração de uma variável segue o formato:

<tipo da variável> <identificador da variável> [=valor] [, ...] ;

Exemplos:

int x;

int y =10;

char um ='1';

char newline = '\n'

boolean verdade;

verdade = true;

float f = 9.1234567f

double d = 9.123456789012345

a) Supondo as instruções:

int x;

int y = 10;

x = 20

resultado = x + y

- Explore como pode escrever no ecrã o texto da linha abaixo:

A soma de x com y é: resultado

Onde está x deve aparecer o valor de x, onde está y deve aparecer o valor de y e onde está resultado deve aparecer o valor da variável resultado.

b) Construa um programa em que teste a declaração de variáveis.

Teste cada um dos tipos primitivos da tabela anterior, **declarando variáveis, atribuindo valores e escrevendo esses valores no ecrã.**

c) Usando os operadores aritméticos que conhece da linguagem C construa expressões simples com variáveis dos tipos int, long, float e double e escreva o resultado no ecrã.

d) As principais instruções de controle (if, for, while) são também iguais às da linguagem C. Para cada uma das alíneas que se seguem, resolva construindo 3 versões: usando um ciclo for; usando um ciclo while; usando um ciclo do ... while.

c1) Escreva os inteiros de 1 a 100.

c2) Escreva os inteiros de 100 a 1.

c3) Calcule o somatório dos inteiros de 1 a 100 e escreva o resultado.

c4) Calcule o produto dos inteiros de 1 a 100 e escreva o resultado.

c5) Calcule o somatório dos inteiros impares de 1 a 100 e escreva o resultado

4 – Operadores

- Construa um programa onde use e teste cada um dos seguintes operadores:

- **aritméticos:** +, -, *, /, %

- **incremento e decremento:** ++, --
(qual a diferença entre `i2=i1++` e `i2=++i1` ?)

- **operadores de atribuição:** +=, -=, *=, /=, %=

5 – Conversão entre tipos

É possível converter um dado tipo num outro compatível (!) usando um operador unário de “casting”. Por exemplo:

```
...  
double x = 2.3;  
int y;  
y = (int) x; /* converte o valor 2.3 no inteiro 2*/  
...  
char c='A';  
int i;  
i = (int) c;
```

- Escreva no ecrã em formato de tabela os caracteres imprimíveis do código ASCII:

| Inteiro | ASCII |
|----------------|--------------|
|----------------|--------------|

| | |
|-----------|-----------------------|
| 32 | -> “espaço em branco” |
|-----------|-----------------------|

| | |
|-----------|------|
| 33 | -> ! |
|-----------|------|

...

| | |
|------------|------|
| 125 | -> } |
|------------|------|

| | |
|------------|------|
| 126 | -> ~ |
|------------|------|