

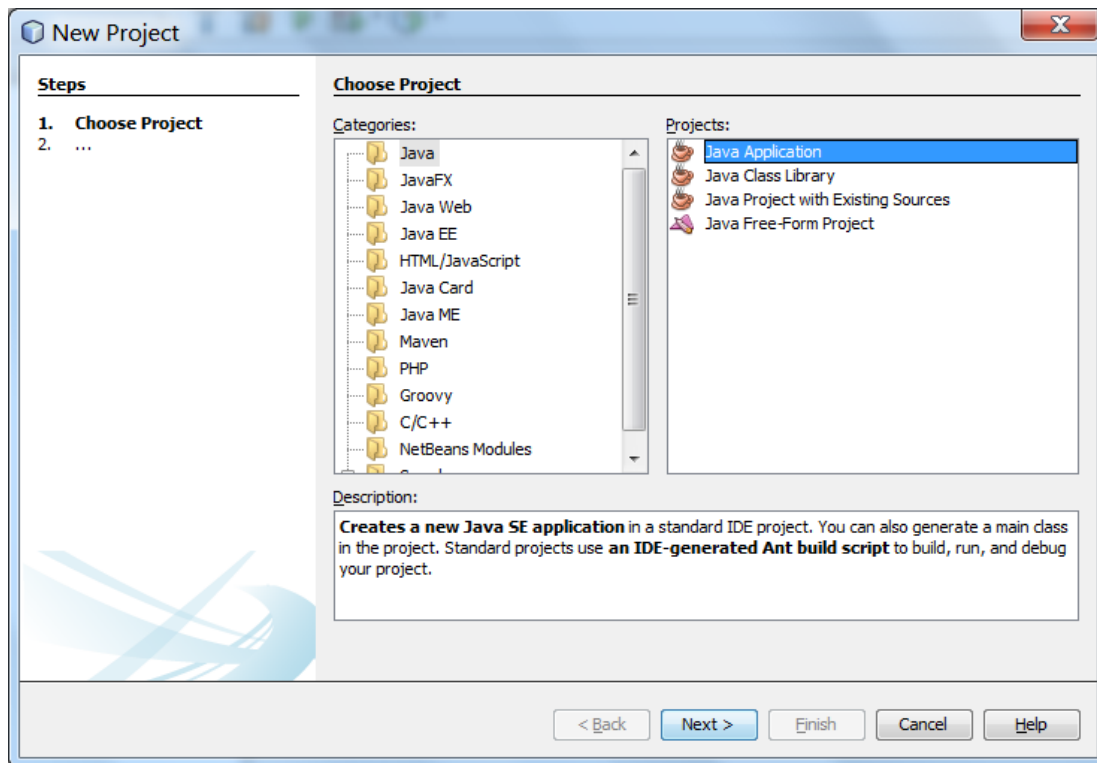
ESQUEMA AULA PRÁTICA 1

- Familiarização com o Ambiente de Desenvolvimento NetBeans
- Introdução à Linguagem de Programação JAVA

0 – Iniciar o ambiente de desenvolvimento integrado (Integrated Development Environment – IDE) NetBeans

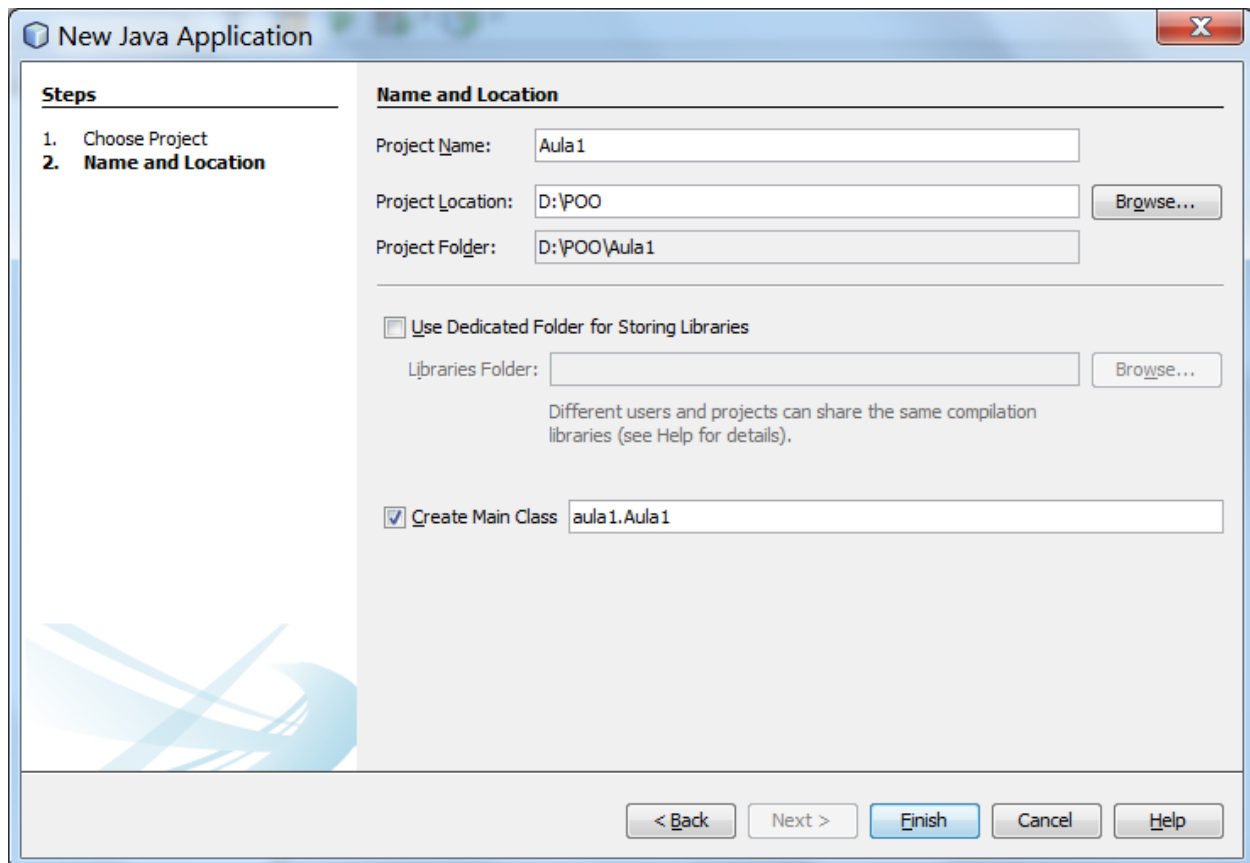
- Start | All Programs | Shortcut to NetBeans

- Inicie o NetBeans e no Menu File selecione a opção New Project. Selecionar a categoria **Java** e o projeto **Java Application**.



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

- 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 **main** pelo qual se inicia a execução do programa.
- Por convenção o nome de uma classe começa por uma letra maiúscula.
- ; é um terminador de instruções.
- { } 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. 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.

Explore a linguagem para cada um dos pontos que se seguem.

2 – Tipos de Dados Primitivos

Tipo	Valores	Valor por omissão	Nº de bits	Gama de valores
boolean	true, false	false	1	--
char	caracteres 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

3 – Valores, variáveis e constantes

Declaração de variáveis:

<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 = 9.1234567

double d = 9.123456789012345

Declaração de constantes:

```
final float pi=3.14159273269;
```

4 - Construa o seguinte programa:

```
class Valores {  
    public static void main(String[] args){  
        int numero;  
        double decNum;  
        numero = -100000;  
        decNum = 12345.6789;  
        System.out.println("O valor da variável inteira é: " + numero);  
        System.out.println("O valor da var. real é: " + decNum);  
        char letra = 'Q';  
        System.out.println( letra);  
    }  
}
```

- Os tipos de dados primitivos da linguagem java são muito parecidos com os da linguagem C que já estudou noutras disciplinas. Experimente no seu primeiro programa declarar variáveis de diferentes tipos, atribuir-lhes um valor, e escrevê-lo no ecrã.

5 – Conversão entre tipos

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

Por exemplo:

```
...  
int x = 2;  
float f;  
f = (float) x; /* converte o valor inteiro de x no real 2.0.*/  
...  
char c='A';  
int i;  
i = (int) c;
```

6 – Ler um carácter do teclado

O método `System.in.read()` permite ler dados a partir de um buffer associado ao teclado.

7 – Construa o seguinte programa

```
class LerCaracter {
    public static void main(String[] args) throws java.io.IOException /* !!! */
    {
        char c;
        System.out.print("Introduza um carácter pelo teclado: ");
        c = (char)System.in.read();
        System.out.println("O carácter que escreveu foi: " + c);
    }
}
```

8 – Estude os seguintes operadores

- aritméticos: +, -, *, /, %
- incremento e decremento: ++, --
(qual a diferença entre `i2=i1++` e `i2=++i1` ?)
- operadores de atribuição: +=, -=, *=, /=, %=

O programa que se segue, está construído numa classe, Segundo, contém uma função `main`, que constitui o código onde terá início a execução, e nessa função são invocadas as outras funções que estão definidas na mesma classe, `getHoras` e `periodoDia`. A classe contém ainda a declaração de um conjunto de variáveis (`saudações`, `nome`, `horas`, `minutos`) que são globais às funções definidas na classe.

Um `GregorianCalendar` é uma classe pré-definida na linguagem Java, que é usada neste programa. Apesar de o programa conter muitos elementos que apenas estudaremos mais para a frente durante este curso, tente explorá-lo.

Para explorar ...

9 – Implemente agora o seguinte programa¹

```
import java.util.*;
public class Segundo {
    private static String[] saudacoes =
        {"Bom dia", "Boa tarde", "Boa noite"};
    private static String nome = "Escreva aqui o seu nome";
    private static int horas;
    private static int minutos;

    public static void getHoras( ){
        GregorianCalendar calend = new GregorianCalendar();
        horas = calend.get(Calendar.HOUR_OF_DAY);
        minutos = calend.get(Calendar.MINUTE);
    }
    private static int periodoDia(int h){
        return (h+20) /8 % 3;
    }
    public static void main(String args[]){
        getHoras ( ) ;
        System.out.println(saudacoes[periodoDia(horas)] + ", " + nome);
        System.out.println("Passam " + minutos + " minutos das " +
            horas + " horas." ) ;
    }
}
```

Depois de analisar o programa, e apesar de ainda não ter conhecimentos para o perceber na totalidade, tente modificá-lo para que:

i) Apresente as horas de um modo gramaticalmente correto (1 hora, 1 minuto! Passam, faltam! Horas exactas?)

ii) Apresente também a data.

iii) Indique o número de dias que faltam até ao fim-de-semana...

Apresente a listagem das datas das 13 próximas Sextas-feiras 13.

¹ Adaptado de “A small cup of Java” de Pedro Guerreiro