

Vitor Amadeu Souza

Implementando um

Perceptron

no

Java

No ambiente IntelliJ IDEA

© 2024 by Cerne Tecnologia e Treinamento Ltda.

© 2024 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Janeiro de 2024

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: Cerne Tecnologia e Treinamento

E-mail da Empresa: cerne@cerne-tec.com.br

Home Page: www.cerne-tec.com.br.com.br

Atendimento ao Consumidor: sac@cerne-tec.com.br

Contato com o Autor: vitor@cerne-tec.com.br



FEITO NO BRASIL

***“Recompensou-me o Senhor conforme a minha justiça,
retribuiu-me conforme a pureza das minhas mãos.”***

SI 18:20

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM, STM etc. Veja os detalhes de um kit na figura abaixo:



Kit Cerne PIC12F

- Microcontrolador PIC12F675
- Botões
- Leds
- Gravação ICSP
- E muito mais!

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais dentre outros. Visite a nossa página na Internet, no endereço www.cerne-tec.com.br e conheça melhor nossos serviços e produtos.



Sumário

Introdução.....	7
Capítulo I – Programação em Java.....	8
1. Ambiente IntelliJ IDEA.....	8
2. Apresentando mensagens.....	11
3. Comentários.....	13
4. Capturando dados no Java.....	15
5. Calculadora I.....	16
6. Calculadora II.....	17
7. Tipos de dados.....	19
8. Operadores da linguagem.....	20
9. Estrutura de repetição for.....	23
10. Estrutura de repetição while.....	28
11. Estrutura de repetição do-while.....	32
12. Declaração de controle if.....	36
Capítulo II – POO – Programação Orientada a Objetos.....	40
1. Classes e objetos.....	40
2. Criação de objetos.....	46
3. Atributos e métodos de classe.....	48
4. Encapsulamento.....	50
5. Métodos.....	55
6. Passagem por valor.....	56
7. Palavra reservada this.....	58
8. Sobrecarga ou overloading.....	59
9. Herança.....	62

10. Classe abstrata.....	66
11. Classe final.....	67
Capítulo III – Interface gráfica.....	69
1. Lendo um nome.....	69
2. Concatenando strings.....	70
3. Calculadora.....	72
4. Caracteres maiúsculos.....	73
Capítulo IV – Interface gráfica.....	76
1. Lendo um nome.....	76
2. Adicionando evento addActionListener.....	77
3. Saudação com caixa de texto.....	79
4. Concatenando strings.....	81
5. Calculadora.....	83
6. Caracteres maiúsculos.....	84
7. Carregando imagens.....	86
8. Carregando imagens 2.....	90
9. Classe gráfica.....	93
10. Combo Box.....	95
11. Option Box.....	97
12. List Box.....	100
13. Alterando cor do form.....	102
14. Calculadora completa.....	105
Capítulo V – Perceptron.....	111
1. Introdução.....	111
2. Implementando o perceptron.....	119

Introdução

A proposta desta literatura é desenvolver a unidade básica usada em Redes Neurais Artificiais (RNA) chamada de Perceptron, usando o Java como ferramenta de programação através de um exemplo prático que demonstra tal implementação.

Para isso, diversos exemplos são abordados apresentando tópicos importantes para se compreender a programação em Java previamente.

A interface de programação usada foi o IntelliJ IDEA versão Community.

Capítulo I

Programação em Java

1. Ambiente IntelliJ IDEA

A proposta deste capítulo é desenvolver diversas experiências na programação em Java. Estes exemplos são importantes, pois permitem um maior entendimento da linguagem Java.

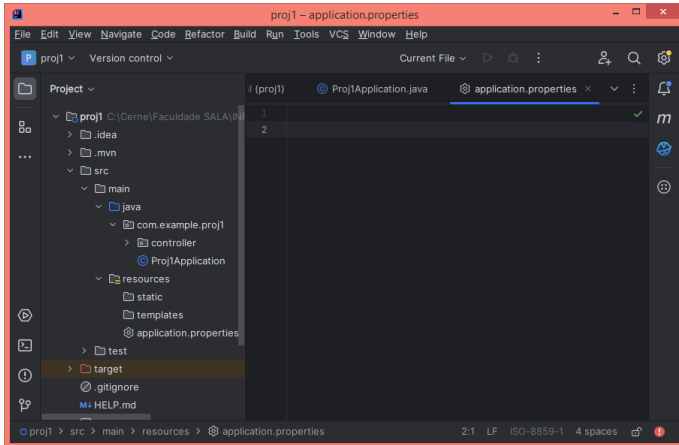
O primeiro passo consiste em baixar a JDK (*Java Development Kit*) de acordo com a versão do seu sistema operacional.

<https://www.oracle.com/br/java/technologies/downloads/>

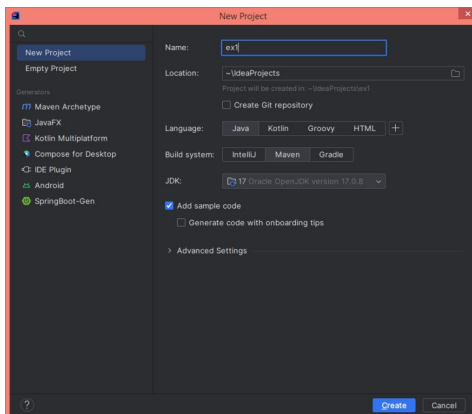
O ambiente de desenvolvimento (IDE) utilizado foi o IntelliJ IDEA, no qual a versão *Community* (gratuita) este pode ser baixado através do link abaixo.

<https://www.jetbrains.com/idea/download/?section=windows>

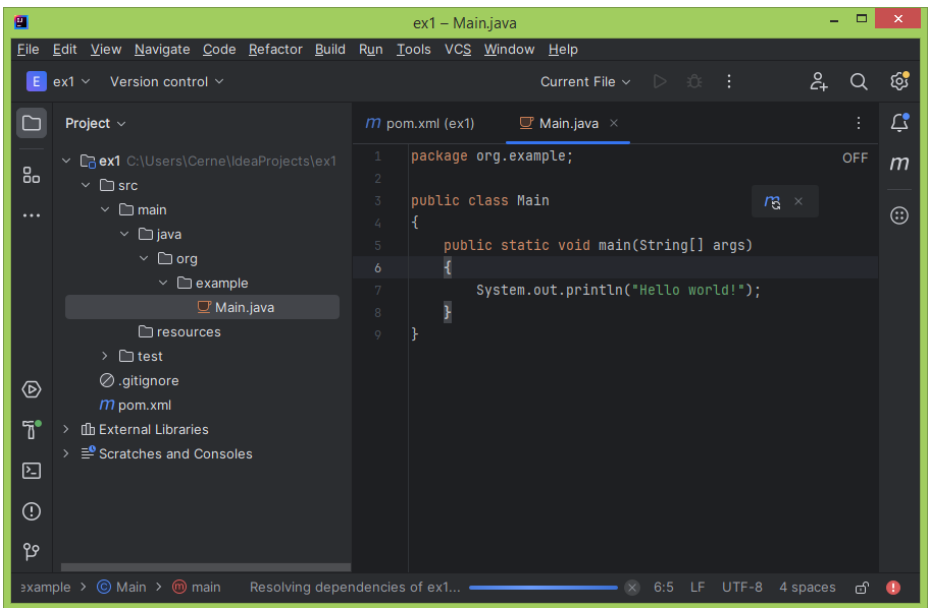
Após o download e instalação, inicialize o IDE. A tela inicial será apresentada na figura abaixo, considerando que a versão utilizada nesta literatura foi a 2023.3.1.




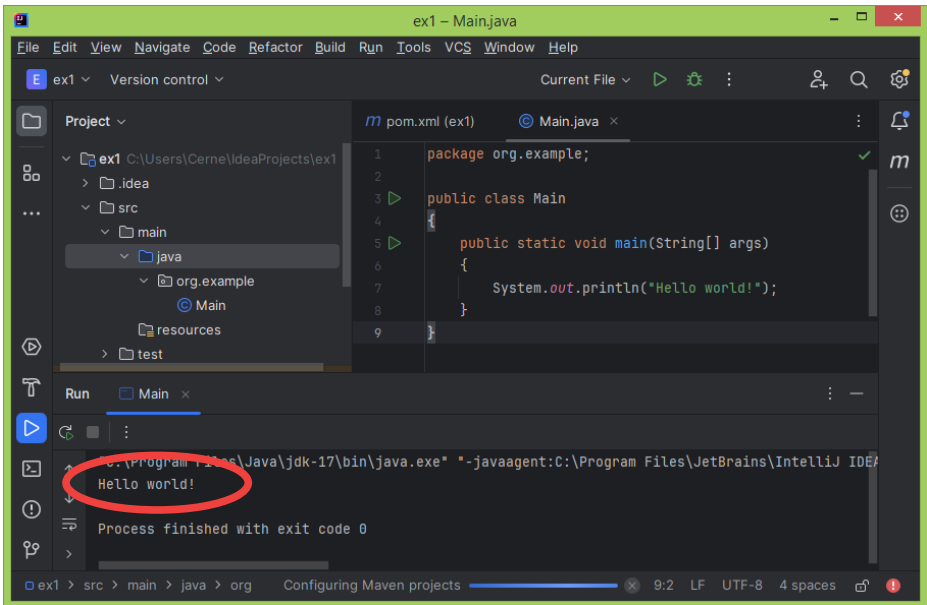
Para criar um projeto, vá ao menu *File-> Project*.



Em *name* escolha o nome do projeto. Em *location*, escolha a pasta no qual o projeto será salvo. Em *language*, escolha Java. Em *build system* escolha *Maven*. Clique no botão *Create*. A tela ficará conforme a imagem abaixo com o novo projeto criado.



O projeto é criado com um código inicial que imprime um valor no console do ambiente. Para rodar o programa, basta clicar no botão  ou ir ao menu *Build* → *Build Project*. Acompanhe o resultado na tela de console.



2. Apresentando mensagens

Digite o código Java abaixo na área de edição.

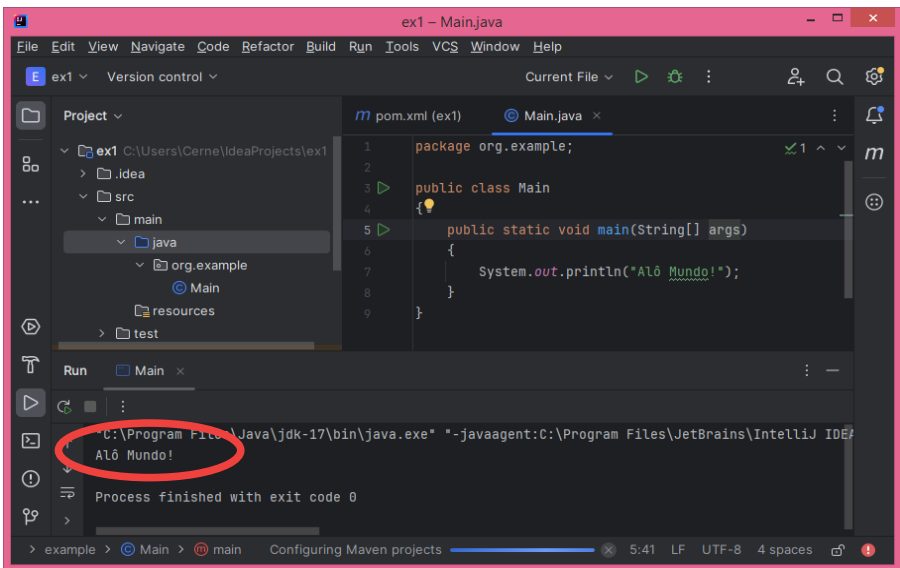
```
package org.example;

public class Main
{
    public static void main(String[] args)
    {
        System.out.println("Alô mundo!");
    }
}
```

O programa inicia com a declaração `public class Main`, onde nesta linha está sendo definida a classe pública que chama-se Main de acordo com o nome do projeto e arquivo criado. A declaração `public static void main(String args[])` define a função principal do programa Java, que é a função main.

A linha `System.out.println("Alô Mundo!")` permite imprimir um valor na tela, no caso a frase “Alô Mundo!”. As chaves marcam o início e fim de cada bloco no Java.

Após escreve este programa, execute-o. Veja abaixo o resultado na janela de Console:



```
ex1 - Main.java
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
ex1 Version control Current File
Project
  ex1 C:\Users\Cerne\IdeaProjects\ex1
    .idea
    src
      main
        java
          org.example
            Main
        resources
        test
Run Main
C:\Program Files\Java\jdk-17\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDE
Alô Mundo!
Process finished with exit code 0
example Main main Configuring Maven projects 5:41 LF UTF-8 4 spaces
```