Universidade Comunitária da Região de Chapecó

Ciência da Computação/Sistemas de informação

2025-2

Disciplina

Apostila 01 – Iniciação

Professor Valdemar LORENZON Junior

Este material é de propriedade intelectual do professor Valdemar Lorenzon Junior e pode ser utilizado se referenciado.

Material original criado em 2001 e reestruturado semestralmente com atualizações necessárias para atender a ementa, atualizações de conteúdo teórico e prático para adaptação ao momento de uso (semestre letivo e atualizações tecnológicas ou acadêmicas).

Sugestões são bem vindas.

juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br

Sumário

O ambiente Java em detalhes de organização de código 4
Workspace 4
Package 4
Import 5
Aplicação 5
Estrutura Básica 6
Componentes Principais 7
Para ter o ambiente IDE VS Code com o Java 7
Exibindo a estrutura de um WKS e projeto de uma aplicação Hello Word em Windows com VS Code. 8
Utilizando o VS Code e organizando o código para criação de programas com classes e aplicação 9

juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br

O ambiente Java em detalhes de organização de código

Conteúdo criado pelo Professor Valdemar Lorenzon Junior através de pesquisa em ferramentas de conteúdo e adaptação e formatação personalizadas.

A pesquisa e formatação de conteúdo visa atualizar o aluno sobre conteúdo a ser utilizado e como forma de ampliação de conceitos e conhecimento.

Na linguagem de programação Java, os conceitos de workspace, package, import e aplicação têm papéis específicos que facilitam a organização, reutilização e manutenção do código. Vamos definir cada um deles:

Workspace

Um Workspace é um conceito principalmente associado a ambientes de desenvolvimento integrados (IDEs), como Eclipse, IntelliJ IDEA, Monorepos, Yarn ou Visual Studio Code.

Ele representa o conjunto de todos os projetos em que você está trabalhando dentro do IDE. Um workspace pode conter um, como vários projetos.

Compartilha-se configurações de ambiente entre estes projetos, assim como personaliza-se configurações por pasta-projeto. Há a ideia de dependências compartilhadas e gerenciamento centralizado.

Um **workspace**, nos ambientes modernos de desenvolvimento, é o espaço lógico ou físico onde você agrupa um ou mais projetos relacionados, com suas configurações, dependências e ferramentas compartilhadas. Ele pode representar um repositório, uma pasta com múltiplos pacotes ou um conjunto de configurações salvas no IDE/editor.

Package

Já falando em Java, um package é um mecanismo de namespace que organiza classes, interfaces, enumerações e anotações relacionadas.

Os packages ajudam a evitar conflitos de nomes e podem ser usados para controlar o acesso, com modificadores de acesso de visibilidade do encapsulamento.

A estrutura de diretórios dos arquivos fonte no sistema de arquivos reflete a organização dos packages, onde cada diretório representa um package. Por exemplo, um arquivo fonte contido no diretório com/minhaempresa/projeto estaria no package com.minhaempresa.projeto.

juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br

Import

A declaração import em Java é usada para permitir o acesso a classes, interfaces, ou outros membros de packages diferentes do package onde a classe atual está definida.

Isso elimina a necessidade de qualificar referências a esses membros com seus nomes de packages completos. Por exemplo, para usar a classe ArrayList que está no package java.util, você pode "importá-la" no início do seu arquivo fonte com import java.util.ArrayList.

Isso permite que você simplesmente use ArrayList em vez de java.util.ArrayList em todo o seu código.

Aplicação

Uma aplicação em Java é um programa que executa em uma máquina virtual Java (JVM). Uma aplicação típica é composta por várias classes que trabalham juntas para realizar determinadas tarefas. Uma aplicação Java pode ser uma aplicação de console, uma interface gráfica de usuário (GUI), uma aplicação web, um serviço web, ou qualquer outro tipo de programa que pode ser desenvolvido usando a linguagem Java. Uma aplicação Java inicia sua execução a partir de um método denominado **main**, que é o ponto de partida da execução do programa.

```
public class App {
    public static void main(String[] args) {
        //
    }
}
```

Esses conceitos desempenham papeis vitais no desenvolvimento de aplicações Java, ajudando os desenvolvedores a organizar, reutilizar e gerenciar o código de maneira eficiente.

A estrutura de um workspace em Java é um conceito organizacional usado principalmente por ambientes de desenvolvimento integrados (IDEs), como Eclipse, IntelliJ IDEA, e Visual Studio Code, para gerenciar projetos de software.

Um workspace é essencialmente um diretório no sistema de arquivos que contém um ou mais projetos Java, juntamente com metadados específicos da IDE que ajudam a gerenciar esses projetos. A estrutura exata pode variar dependendo da IDE utilizada, mas geralmente segue um padrão comum.

juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br

Estrutura Básica

Um workspace típico em Java pode ter a seguinte estrutura de diretórios e arquivos (Baseado no VS Code):

```
workspace/
|
├── projetoA/
|   ├── .vscode/                # Pasta de configurações da IDE do projeto
|   |   |   ├── settings.json   # Arquivo JSON de configurações IDE do projeto
|   ├── src/                    # Pasta de código-fonte
|   |   ├── App.Java            # Classe de aplicação
|   |   ├── (pacote)/           # Pacotes para organização do código
|   |   |   ├── ClasseA.java
|   |   |   └── ClasseB.java
|   |   └── ...
|   ├── bin/                    # Pasta de saída para arquivos compilados
|   |   |   ├── App.class
|   |   ├── (pacote)/
|   |   |   ├── ClasseA.class
|   |   |   └── ClasseB.class
|   |   └── ...
|   ├── lib/                    # Repositório de dependências externas
|   |   └── biblioteca.jar
```


juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br

Componentes Principais

.vscode/ settings.json: Este diretório contém configurações para o workspace como um todo, incluindo informações sobre os projetos contidos nele e preferências específicas da IDE. Podem haver configurações globais, workspace e projeto, tendo então cada qual seu arquivo json particular.

projetoA/, projetoB/, etc.: Estes são diretórios de projetos individuais dentro do workspace. Cada projeto pode conter seu próprio código-fonte, bibliotecas e arquivos de configuração.

src/: A pasta onde o código-fonte do projeto é mantido. Dentro dela, o código é organizado em pacotes, que são diretórios que correspondem ao namespace do código.

lib/: Contém as bibliotecas (arquivos .jar e outros) que o projeto depende.

bin/: Diretório de saída padrão para os arquivos .class compilados pelo IDE a partir do código-fonte.

Para ter o ambiente IDE VS Code com o Java

Para criar e configurar o ambiente IDE VS Code com o Java é necessário efetuar:

1. Instalar o Java (JDK);

   - Baixe o JDK recomendado: jdk.java.net;
   - Recomendado: JDK 17 (versão LTS mais recente e estável, mínimo 17);

2. Instalar o VS Code

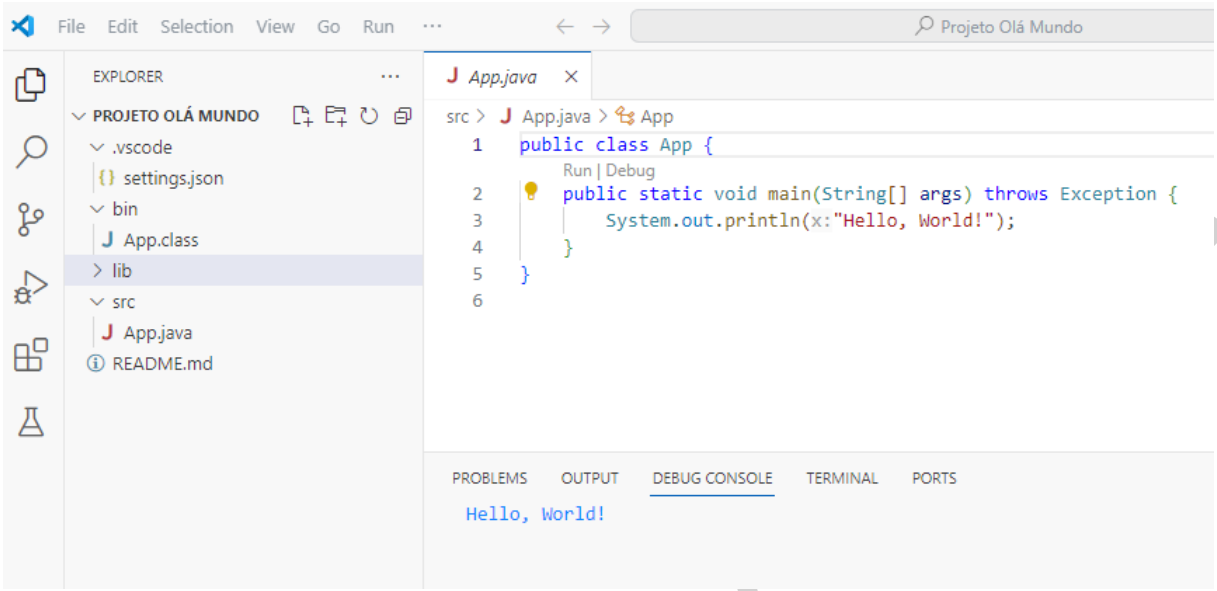
   - Baixe em: code.visualstudio.com;
   - Após instalar, abra o VS Code;

3. Instalar as extensões Java no VS Code: (Ctrl+Shift+X) (Microsoft);

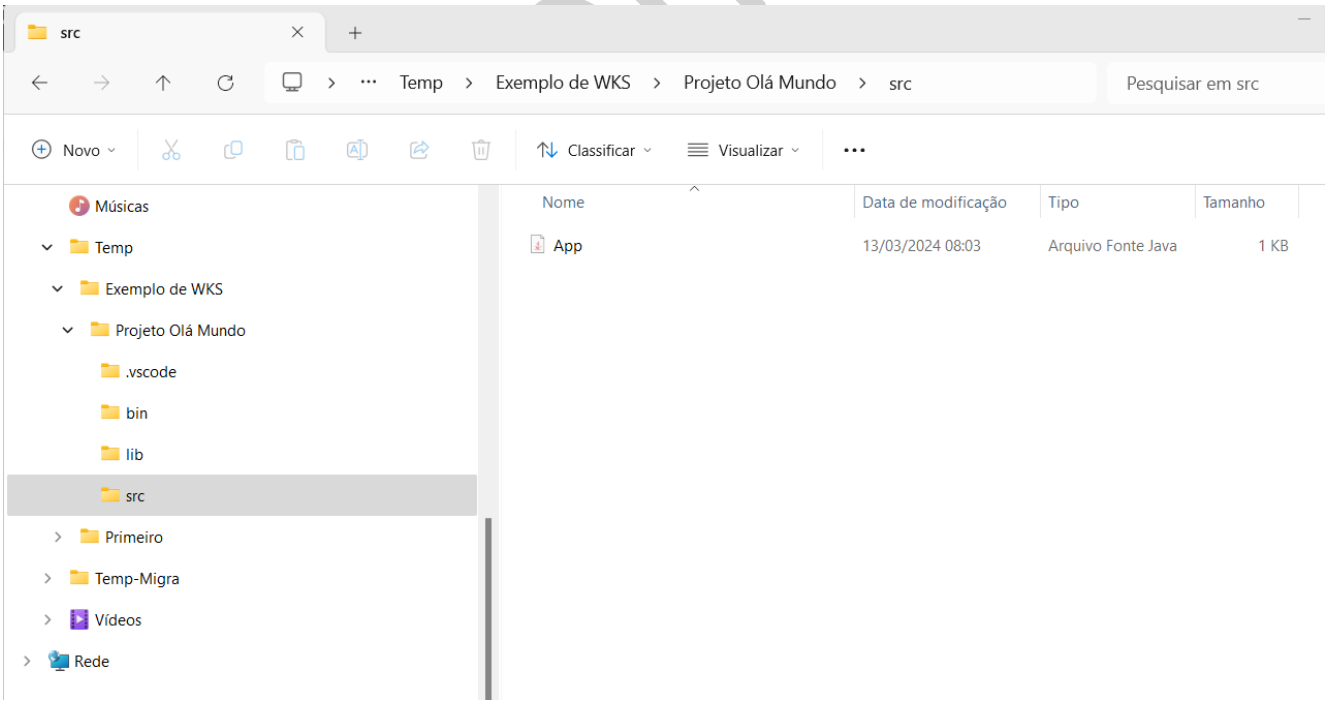
   - Extension Pack for Java (Microsoft);
   - Project Manager for Java (Microsoft);
   - Debugger for Java (Microsoft);

juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br

Exibindo a estrutura de um WKS e projeto de uma aplicação Hello Word em Windows com VS Code.



Pelo Explorer:



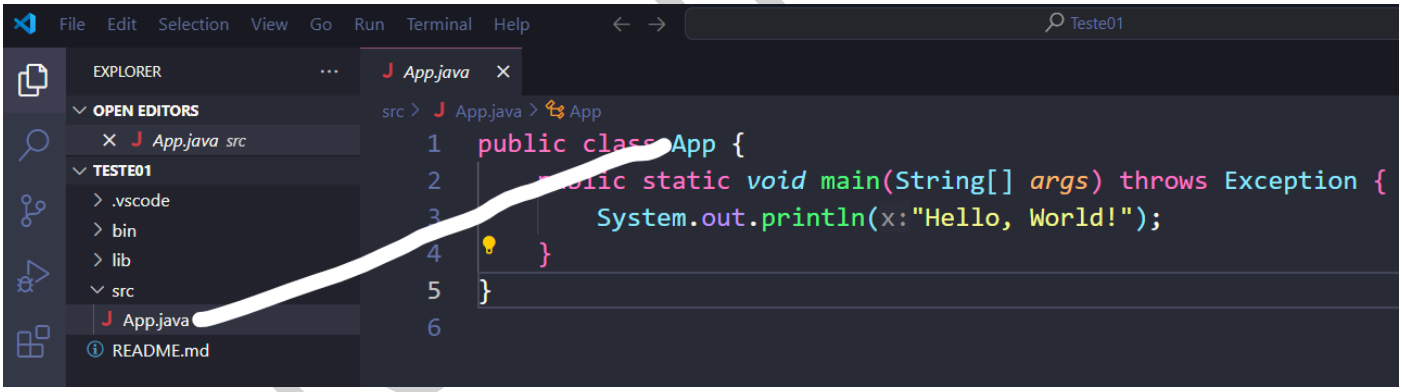
juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br

Utilizando o VS Code e organizando o código para criação de programas com classes e aplicação

Segue, passo a passo em tópicos apresentados na disciplina de como organizar seus programas em Java com o uso de VSCode e de acordo com a expectativa de uso inicial e contínuo na disciplina:

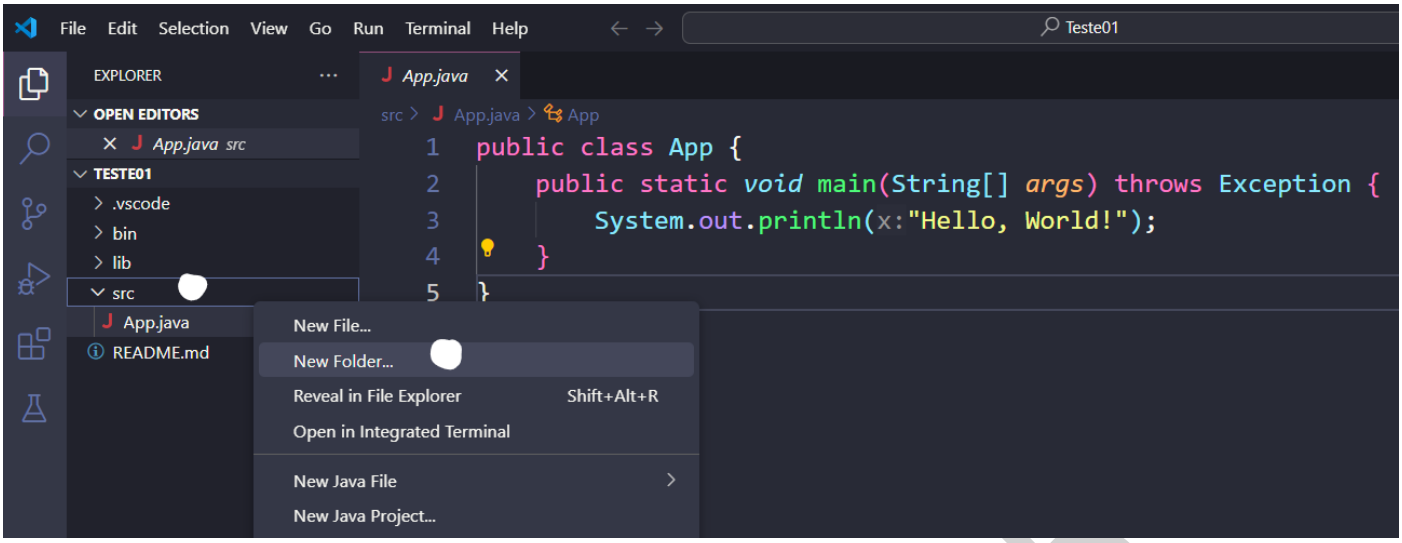
1 – Ao criar um novo projeto a sugestão de classe de aplicação é apresentada, nomeie a sua classe e programa .java pré criado de acordo com a sua especificação. Utilizando CammelCase Uppercase;

a) Para criar um novo projeto comece com File, New Window;
b) A seguir, New File...
c) Escolha New Java Project e No build tools;
d) A seguir escolha o workspace da aplicação;
e) Crie com indicação do nome solicitado a pasta onde ficará a aplicação;
f) Uma classe e arquivo App serão criados nesta pasta;
g) Renomeie o arquivo da aplicação e classe;
    a. Sua classe de aplicação e seu .java dentro do src devem conter este mesmo nome.



2 – Crie uma pasta "**`utils`**" no src da sua aplicação, será um pacote de utilitários;

juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br



3 – Deposite a classe "Teclado" pelo arquivo Teclado.java nesta pasta utils;

4 – Importe a classe Teclado na sua aplicação com o uso de **`import utils.Teclado`**;

5 – Ao criar classes, crie um novo arquivo junto com a sua classe de aplicação e salve o mesmo com o nome da classe usando a notação CammelCase Uppercase e no "pacote" src. Sendo assim, não será necessário "import" da classe na aplicação. Desta forma, no src estará a sua aplicação e qualquer classe criada separadamente e unicamente uma classe por arquivo.

a) Para criar um novo arquivo, clique com o botão direito do mouse na pasta que quer criar o arquivo e escolha "New Java file...", "New class..."

b) Nomeie o arquivo conforme indicado.

Como "abrir" projetos já existentes?

De posse de uma estrutura Vs Code já salva em um local, use File..., Open folder... e selecione e abra a pasta do projeto salvo.

Exemplo serão os arquivos repassados pelo professor para os primeiros passos com a linguagem Java. Descompacte os arquivos e, abra as pastas de cada projeto conforme for estudar.

juniorlorenzonvaldemar@gmail.com vljunior@unochapeco.edu.br