INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BRASÍLIA

TECNÓLOGO SISTEMAS PARA INTERNET

Professor: Gustavo Henrique Dornelas de Deus

Resumão – Programação Orientada a Objetos

Data de disponibilização do trabalho no Moodle, pelo professor: 14/04/2022

Data de entrega da tarefa pronta, pelo aluno: 19/05/2022

Valor: 2,00

Atenção: Trabalhos entregues fora da data só serão aceitos mediante apresentação de atestamento médico.

Identificação

Nome: Paulo Estevão Santiago Teixeira

Para cada um dos conceitos, procure 2 autores que os contextualizam observando sempre a aderência do texto a disciplina de Programação Orientada a Objetos, informando o livro que foi retirado a citação e a referida página.

1 – POO - Programação Orientada a Objetos

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Marjin Haverbake	**2018**	**149**	**Eloquent javascript, 3rd Edition**	**"***In programming culture, we have a thing called object-oriented programming, a set of techniques that use objects (and related concepts) as the central principle of program organization. Though no one really agrees on its precise definition, object-oriented programming has shaped the design of many programming languages, including JavaScript.***"**
Herbert Schildt	**2014**	**17**	**Java: The complete reference, 9rd edition**	*" Object-oriented programming organizes a program around its data (that is, objects) and a set of well-defined interfaces to that data. An object-oriented program can be characterized as data controlling access to code."*

2 – Abstração

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Marjin Haverbake	**2018**	**127**	**Eloquent javascript,3rd Edition**	**"***Abstractions hide details and give us the ability to talk about problems at a higher (or more abstract) level.***"**
Herbert Schildt	**2014**	**18**	**Java: The complete reference, 9rd edition**	*"An essential element of object-oriented programming is abstraction. Humans manage complexity through abstraction. For example, people do not think of a car as a set of tens of thousands of individual parts. They think of it as a well-defined object with its own unique behavior. This abstraction allows people to use a car to drive to the grocery store without being overwhelmed by the complexity of the parts that form the car. They can ignore the details of how the engine, transmission, and braking systems work. Instead, they are free to utilize the object as a whole"*

3 - Encapsulamento

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Marjin Haverbake	**2018**	**150**	**Eloquent javascript,3rd Edition**	**"***Separating interface from implementation is a great idea. It is usually called encapsulation.***"**
Herbert Schildt	**2014**	**18**	**Java: The complete reference, 9rd edition**	**"***Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse. One way to think about encapsulation is as a protective wrapper that prevents the code and data from being arbitrarily accessed by other code defined outside the wrapper. Access to the code and data inside the wrapper is tightly controlled through a well-defined interface.***"**

4 – Polimorfismo

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Marjin Haverbake	**2018**	**164**	**Eloquent javascript,3rd Edition**	**"***When a piece of code is written to work with objects that have a certain interface, any kind of object that happens to support this interface can be plugged into the code, and it will just work. This technique is called polymorphism. Polymorphic code can work with values of different shapes, as long as they support the interface it expects.***"**
Herbert Schildt	**2014**	**22**	**Java: The complete reference, 9rd edition**	**"***More generally, the concept of polymorphism is often expressed by the phrase "one interface, multiple methods." This means that it is possible to design a generic interface to a group of related activities. This helps reduce complexity by allowing the same interface to be used to specify a general class of action. It is the compiler's job to select the specific action (that is, method) as it applies to each situation. You, the programmer, do not need to make this selection manually***"**

5 – Herança

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Marjin Haverbake	**2018**	**174**	**Eloquent javascript,3rd Edition**	**"***Inheritance allows us to build slightly different data types from existing data types with relatively little work. It is a fundamental part of the object-oriented tradition***"**
Herbert Schildt	**2014**	**161**	**Java: The complete reference, 9rd edition**	**"***Inheritance is one of the cornerstones of object-oriented programming because it allows the creation of hierarchical classifications. Using inheritance, you can create a general class that defines traits common to a set of related items. This class can then be inherited by other, more specific classes, each adding those things that are unique to it. In the terminology of Java, a class that is inherited is called a superclass. The class that does the inheriting is called a subclass. Therefore, a subclass is a specialized version of a superclass. It inherits all of the members defined by the superclass and adds its own, unique elements.***"**

6 – Classes abstratas

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Tarun Telang	2021		https://www.educative.io/edpresso/what-are-abstract-classes-in-java	"*An `abstract` class is a special type of class that cannot be instantiated. It contains one or more `abstract` methods, which are methods with unimplemented code. It is used to represent pure concepts without any detail.*"
Herbert Schildt	2014	182	Java: The complete reference, 9rd edition	*"Any class that contains one or more abstract methods must also be declared abstract. To declare a class abstract, you simply use the abstract keyword in front of the class keyword at the beginning of the class declaration. There can be no objects of an abstract class. That is, an abstract class cannot be directly instantiated with the new operator. Such objects would be useless, because an abstract class is not fully defined. Also, you cannot declare abstract constructors, or abstract static methods. Any subclass of an abstract class must either implement all of the abstract methods in the superclass, or be declared abstract itself"*

7 - Interface

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Paul Deitel, Harvey Deitel	2011	284	Java: how to program, 9rd edition	"*An interface declaration begins with the keyword interface and contains only constants and abstract methods. Unlike classes, all interface members must be public, and interfaces may not specify any implementation details, such as concrete method declarations and instance variables. All methods declared in an interface are implicitly public abstract methods, and all fields are implicitly public, static and final.* "
Herbert Schildt	2014	196	Java: The complete reference, 9rd edition	"*Using the keyword interface, you can fully abstract a class' interface from its implementation. That is, using interface, you can specify what a class must do, but not how it does it. Interfaces are syntactically similar to classes, but they lack instance variables, and, as a general rule, their methods are declared without any body.*

8 – Coleções de objetos

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Paul Deitel, **Harvey Deitel**	**2011**	**284**	**Java: how to program, 9rd edition**	*"The Java API provides several predefined data structures, called collections, used to store groups of related objects. These classes provide efficient methods that organize, store and retrieve your data without requiring knowledge of how the data is being stored. This reduces application-development time."*
Herbert Schildt	**2014**	**499**	**Java: The complete reference, 9rd edition**	**"***"The Collections Framework was designed to meet several goals. First, the framework had to be high-performance. The implementations for the fundamental collections (dynamic arrays, linked lists, trees, and hash tables) are highly efficient. You seldom, if ever, need to code one of these "data engines" manually. Second, the framework had to allow different types of collections to work in a similar manner and with a high degree of interoperability. Third, extending and/or adapting a collection had to be easy. Toward this end, the entire Collections Framework is built upon a set of standard interfaces. Several standard implementations (such as LinkedList, HashSet, and TreeSet) of these interfaces are provided that you may use as-is. You may also implement your own collection, if you choose.***"**"

9 – Tratamento de exceções

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Paul Deitel, **Harvey Deitel**	**2011**	**439**	**Java: how to program, 9rd edition**	**"***Exception handling enables you to create applications that can resolve (or handle) exceptions. In many cases, handling an exception allows a program to continue executing as if no problem had been encountered. The features presented in this chapter help you write robust and fault-tolerant programs that can deal with problems and continue executing or terminate gracefully. Java exception handling is based in part on the work of Andrew Koenig and Bjarne Stroustrup***"**
Herbert Schildt	**2014**	**213**	**Java: The complete reference, 9rd edition**	**"***A Java exception is an object that describes an exceptional (that is, error) condition that has occurred in a piece of code. When an exceptional condition arises, an object representing that exception is created and thrown in the method that caused the error. That method may choose to handle the exception itself, or pass it on. Either way, at some point, the exception is caught and processed. Exceptions can be generated by the Java run-time system, or they can be manually generated by your code. Exceptions thrown by Java relate to fundamental errors that violate the rules of the Java language or the constraints of the Java execution environment. Manually generated exceptions are typically used to report some error condition to the caller of a method. Java exception handling is managed via five keywords: try, catch, throw, throws, and finally.*

10 – MVC – Model View Controller

Autor(es)	Ano	Página	Nome do Livro/Artigo	Citação
Herbert Schildt	**2014**	**1213**	**Java: The complete reference, 9rd edition**	**"***No matter what architecture is used to implement a component, it must implicitly contain these three parts. Over the years, one component architecture has proven itself to be exceptionally effective: Model-View-Controller, or MVC for short. The MVC architecture is successful because each piece of the design corresponds to an aspect of a component. In*

				MVC terminology, the model corresponds to the state information associated with the component. For example, in the case of a check box, the model contains a field that indicates if the box is checked or unchecked. The view determines how the component is displayed on the screen, including any aspects of the view that are affected by the current state of the model. The controller determines how the component reacts to the user. For example, when the user clicks a check box, the controller reacts by changing the model to reflect the user's choice (checked or unchecked). This then results in the view being updated. By separating a component into a model, a view, and a controller, the specific implementation of each can be changed without affecting the other two. For instance, different view implementations can render the same component in different ways without affecting the model or the controller"
Zanfina Svirca	**2020**	https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1	**Everything you need to know about MVC architecture**	"*MVC is known as an architectural pattern, which embodies three parts Model, View and Controller, or to be more exact it divides the application into three logical parts: the model part, the view and the controller. It was used for desktop graphical user interfaces but nowadays is used in designing mobile apps and web apps*."