|  |
| --- |
| Phiên bản 1.0.0 |
|  |
| Tài liệu Java Basic |
|  |
|  |
|  |
|  |
|  |
| Người lập tài liệu: Khổng Minh Mạnh |

# **Danh mục thay đổi tài liệu**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Ngày thay đổi** | **Người thay đổi** | **Nội dung thay đổi** |
| 1 | 14/05/2022 | ManhKM | Tạo mới tài liệu |
| 2 |  |  |  |

Bài 1: Hướng dẫn cài đặt môi trường + thiết lập môi trường

Bài 2: Hướng dẫn tạo dự án Java đầu tiền

Bài 3: Hướng dẫn phát hành dự án Java

Lý thuyết:

Việc phát hành sản phẩm hay việc cài cắm ứng dụng Java lên các máy chủ.

Thông qua việc build ra các gói jar, war, hoặc các file class.

Trong đó:

Chỉ định ra một class là class chạy đầu viên của Chương trình -> Main Class.

Thực hành:

Build project bằng cách: java -jar xxx.jar

Bài 4: Khai báo biến & toán tử trong Java

+ Toán tử điều kiện xxx?xxx:

Bài 5: Tìm hiểu về If, else, switch trong lập trình Java

+ Tìm hiểu về lớp Scanner

Cách khai báo, cách sử dụng.

Một trong các tính năng của java.utils.\*

Bài 6: Tìm hiểu Loop (for, while, do while) trong Java

Bài 7: Tìm hiểu về mảng Array và List trong Java

Bài 8: Hướng dẫn Collection Java | ArrayList Java | Vector Java | Map & HashMap Java | String Java

+ Khi khai báo mảng -> Array cần phải có số phần tử trong mảng trước.

Giá trước của mảng phải được xác định trước khi tạo mảng.

+ Hướng dẫn Collection trong Java

+ Interface có kế thừa hay không?

Interface extend Interface

List<String> list = new ArrayList<>();

List chính là một Interface

ArrayList là một Object

Đây chính là một tính chất kế thừa của Interface.

+ Các thao tác liên quan đến CRUD Database.

+ Các thao tác duyệt qua các phần tử trong mảng.

For | for each | while | …

+ Thành phần Vector

Nó được kế thừa từ List

+ Bài tập:

Bài 1: Tìm chuỗi trong chuỗi

Graphical user interface, text, application

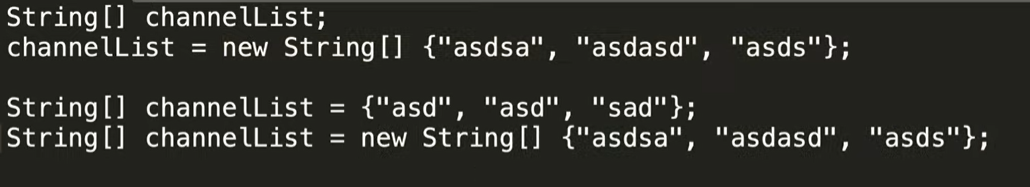
Description automatically generated

Hướng dẫn cách tối ưu source:

Graphical user interface, text, chat or text message

Description automatically generated

Bài 9: Giải thuật tìm kiếm chuỗi Java | Tìm hiểu Exception Java cho người mới bắt đầu



[Tìm kiếm chuỗi trong chuỗi + Exception] <https://gokisoft.com/share-code-tim-kiem-chuoi-trong-chuoi--exception.html>

[Test tìm kiếm chuỗi trong chuỗi] <https://gokisoft.com/test-tim-kiem-chuoi-trong-chuoi.html>

+ Kiểu truy xuất dữ liệu trong Java -> có bao nhiêu cách và cách dùng như nào: public, protected, default, private.

+ Các kiểu Exception trong Java.

Tìm cách giải quyết trong trường hợp có Exception.

Bài 10: Tìm hiểu về Class và Object trong lập trình Java

+ Lý thuyết:

Việc quản lý thông tin liên quan đến các Class -> hình tượng hóa thành các Object.

Thông qua việc quản lý các class thì đó sẽ là quản lý luôn các Object.

Class mục đích để sinh ra giúp dev tạo ra một file -> quản lý các Object thông qua Class đó.

+ Lý thuyết về hàm tạo trong một class -> dùng để tạo một đối tượng -> hàm constructor

+ Lý thuyết liên quan đến this -> nó là chỉ chính đối tượng đó và trỏ đến các thuộc tính trong chính đối tượng đó.

+ Lý thuyết về Access Modifier: private, default, protected, public.

Diagram

Description automatically generated

Hình: Access Specifiers trong Java

Table

Description automatically generated

Bài 11: Tìm hiểu OOP trong Java cho người mới bắt đầu

+ Lý thuyết

Lập trình hướng đối tượng là gì?

Các tính chất trong lập trình hướng đối tượng

Tính bao đóng: (Today)

Tính kế thừa

Tính đa hình

Tình trừu tượng

++++

Xác định -> việc triển khai code như nào -> flow chính trong Project.

Xác định -> có những đối tượng nào

Suy luận đưa ra thành các đối tượng cụ thể.

Ví dụ:

Sinh viên -> thì sẽ quản lý những đối tượng gì trong sinh viên này.

Những thuộc tính của sinh viên (properties): Tên, tuổi, địa chỉ, email, sdt, rollno

Những hành động của sinh viên (Function, hàm, method, phương thức): Học, chạy bộ, …

Khái báo tất cả các thuộc tính + hành động -> lớp đối tượng OOP.

Triển khai code như thế nào?

+ Coding ví dụ: Quản lý Student

+ Các tính tổng quát trong lập trình hướng đối tượng.

Khi nói đến Student -> các thuộc tính và hành động -> khi viết các tính chất trong đối tượng Student thì nó gọi là bao đóng.

+ Tính chất kế thừa:

Ví dụ: Quản lý công dân trong thành phố hà nội -> Citizen + Sinh viên + Engineer + …

+ Khái niệm Overriding, Overloading:

Nếu không có từ khóa @Overriding, @Overloading thì nó sẽ có warning cảnh báo -> sẽ thông báo khi cần thay đổi giá trị nào đó trong method Overide hay Overload.

+ Khái niệm Setter/Getter

Bài 12: Kế thừa trong OOP – Lập trình Java

Diagram

Description automatically generated

Hình: Mô tả các quan hệ giải thích trong OOP: HinhHoc, HCN, TamGiac

Graphical user interface, text, application

Description automatically generated

Giải thích tính đa hình thông qua ví dụ kiểu này.

Giải thích 2 khung thời gian khác nhau.

Tại lúc viết code -> thì hh là đại diện cho Lớp Hình Học

Tại thời điểm biên dịch + chạy code -> hh lại là được khởi tạo từ chính TamGiac()

Note: Tại sao phải tạo Đối tượng Abstract từ một Đối tượng không Abstract -> Tính đa hình được hiểu là gì?

Trong quá trình phát triển dự án có nhiều các lớp khác nhau -> Thì sử dụng Abstract | Interface dùng để chỉ tương tác đến các thành phần chung thôi -> lúc này nó khá là tiện lợi.

Bài 13: Tìm hiểu OOP Java Cho người bắt đầu

+ Lý thuyết

Phân biệt Overriding và Overloading trong Java

TL: Phân biệt nhau qua tham số truyền vào.

+ Mục đích của việc sử dụng bộ Setter/Getter:

Sử dụng để validate giá trị đối với Object.

+ Giải thích về đặc tính đa hình

Nếu trong class con có thêm thuộc tính so với class cha -> thì có thể tạo ra hàm tạo rồi call đến super class constructor của lớp cha, phần còn lại không có thì thông qua this.color = color.

+ Thời gian giải thích chi tiết: Phút 25 đến …