

+ Java Cơ Bản

- Java là gì?
- Lịch sử Java
- Các tính năng của Java
- Cài đặt môi trường Java (cho người mới bắt đầu)
- Thiết lập PATH cho java
- Chương trình Hello World
- JDK, JRE và JVM
- Biến trong java
- Các kiểu dữ liệu trong java
- Ép kiểu (Type casting)
- Các toán tử trong Java
- Hệ thống Unicode trong java

+ Câu Lệnh Điều Khiển

- Mệnh đề If-else
- Mệnh đề Switch-case
- Vòng lặp For
- Vòng lặp While
- Vòng lặp Do-While
- Break trong java
- Continue trong java
- Chương trình java kinh điển

+ Các Tính Chất Của Java

OOPs

- Tính đóng gói trong java
- Tính kế thừa trong java
- Tính đa hình trong java
- Nạp chồng phương thức
- Ghi đè phương thức
- Overloading vs Overriding

+ Các Khái Niệm Java

OOPs

- Tổng quan java OOPs
- Lớp và đối tượng
- Package trong java
- Constructor trong java
- Từ khóa this trong java
- Từ khóa super trong java
- Từ khóa final trong java
- Từ khóa static trong java
- Access Modifier trong java
- Abstract class trong java
- Interface trong java



BẢN QUYỀN
kaspersky

KẾT NỐI AN TOÀN
KHÔNG CÒN VIRUS

XEM NGAY

Bộ nhớ stack và heap trong java

Học Java cơ bản

Immutable String trong java

Một sự hiểu biết toàn diện về bộ nhớ stack và heap trong java là rất quan trọng để hiểu được một chương trình hoạt động như thế nào. Các khái niệm này cung cấp một khuôn khổ cho việc hiểu ứng dụng hoạt động như thế nào và là cơ sở cho việc thực hiện hệ thống runtime được sử dụng trong Java.

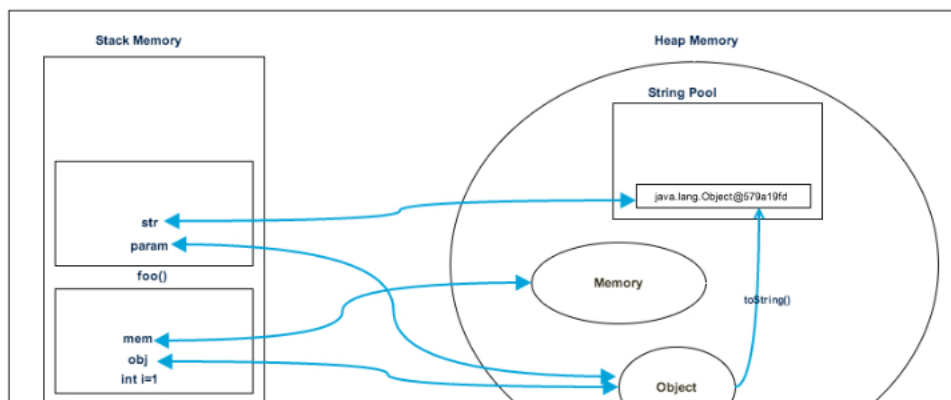
Nội dung chính:

Khái niệm về bộ nhớ stack và heap trong java:
Ví dụ về bộ nhớ stack và heap trong java

Khái niệm về bộ nhớ stack và heap trong java:

Stack là một vùng nhớ được sử dụng để lưu trữ các tham số và các biến local của phương thức mỗi khi một phương thức được gọi ra. Các tham số và các biến local của một phương thức tạo thành một bản ghi kích hoạt, còn được gọi là một **stack frame**. Các bản ghi kích hoạt được đẩy vào một stack khi phương thức được gọi và đẩy ra khỏi stack khi phương thức trả về. Sự tồn tại tạm thời của các biến này quyết định thời gian sống của các biến.

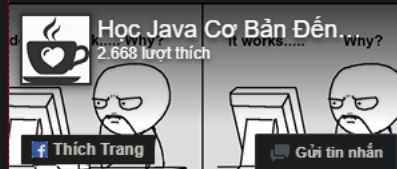
Heap là một vùng nhớ trong bộ nhớ được sử dụng để lưu trữ các đối tượng khi từ khóa new được gọi ra, các biến static và các biến toàn cục (biến instance).



Recent Updates

- Đọc file CSV trong Python
- Lệnh DELETE MySQL trong Python
- Lệnh UPDATE MySQL trong Python
- Lệnh SELECT MySQL trong Python
- Lệnh INSERT MySQL trong Python
- Cài đặt môi trường MySQL cho Python
- Tạo bảng MySQL trong Python
- Tạo new database MySQL trong Python
- Kết nối Python với MySQL
- Truyền tương dữ liệu trong Python
- Ghi đè phương thức trong Python - Method Overriding
- Kế thừa trong Python

VietTuts on facebook



1 người bạn thích nội dung này



- ☑️ Interface trong java
- ☑️ Abstract vs Interface trong java
- ☑️ Lớp Object trong java
- ☑️ Object Cloning trong java
- ☑️ equals() và hashCode()
- ☑️ Array (mảng) trong java
- ☑️ Khai báo mảng trong java
- ☑️ Duyệt mảng trong java
- ☑️ Lớp Wrapper trong java
- ☑️ Đề quy trong java
- ☑️ Truyền giá trị và tham chiếu trong java
- ☑️ Toán tử instanceof trong java
- ☑️ Sử dụng regex trong java

+ Java String

+ Xử Lý Ngoại Lệ

+ Các Lớp Lồng Nhau

+ Đa Luồng (Multithreading)

+ Java AWT

+ Java Swing

+ Java I/O

+ Ví Dụ Java I/O

+ Lập Trình Mạng Với Java

+ Java Date

+ Chuyển Đổi Kiểu Dữ Liệu

+ Java Collections

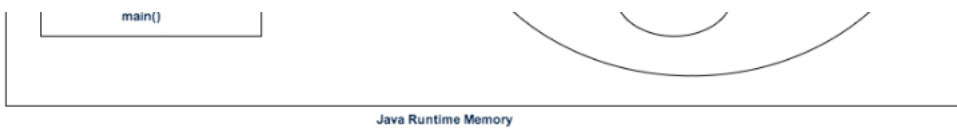
+ Java JDBC

+ Các Tính Năng Mới Trong Java

+ Java Math

+ Bài Tập Java Có Lời Giải

+ Câu Hỏi Phỏng Vấn Java



Bộ nhớ Stack tăng lên và tiến về phía heap khi một phương thức được gọi và co lại khi phương thức trả về. Heap không phát triển theo một trình tự dự đoán được và có thể trở nên phân mảnh. Khi chúng chia sẻ cùng một không gian bộ nhớ, nếu heap và stack va chạm thì chương trình sẽ bị chấm dứt.



Việc hiểu biết về stack và heap là quan trọng bởi vì:

- Nó cung cấp một kiến thức cơ sở để hiểu về data được tổ chức như thế nào trong một ứng dụng.
- Nó củng cố khái niệm về phạm vi và thời gian sống của một biến.
- Nó giúp cho việc giải thích về đệ quy làm việc như thế nào.

Ví dụ về bộ nhớ stack và heap trong java

File: Customer.java

```
import java.math.BigDecimal;
import java.util.Locale;

public class Customer {
    private String name;
    private int accountNumber;
    private Locale locale;
    private BigDecimal balance;

    public Customer() {
        this.name = "Default Customer";
        this.accountNumber = 12345;
        this.locale = Locale.ENGLISH;
        this.balance = new BigDecimal("0");
    }

    public String getName() {
        return name;
    }

    public void setName(String name) throws Exception {
        if (name == null) {
            throw new Exception("Names must not be null");
        } else {
            this.name = name;
        }
    }

    public int getAccountNumber() {
        return accountNumber;
    }

    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    public BigDecimal getBalance() {
        return balance;
    }
}
```





```

public void setBalance(float balance) {
    this.balance = new BigDecimal(balance);
}

public String toString() {
    java.text.NumberFormat format;
    format = java.text.NumberFormat.getCurrencyInstance(locale);
    return format.format(balance);
}
}

```

File: CustomerDriver.java

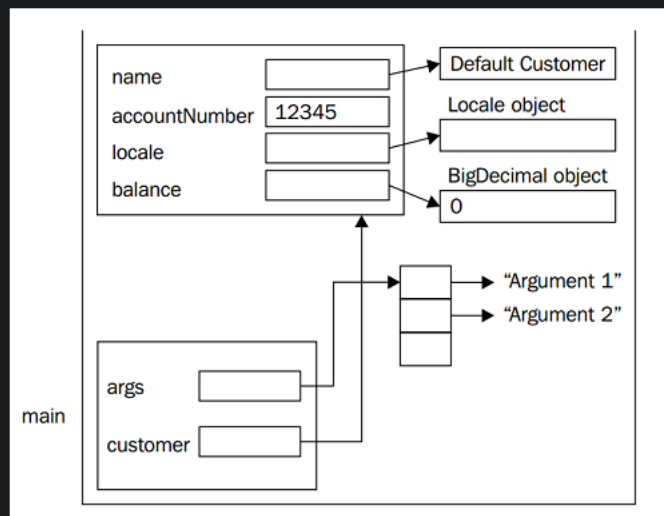
```

public class CustomerDriver {
    public static void main(String[] args) {
        Customer customer; // defines a reference to a Customer
        customer = new Customer(); // Creates a new Customer object
        customer.setBalance(12506.45f);
        System.out.println(customer.toString());
    }
}

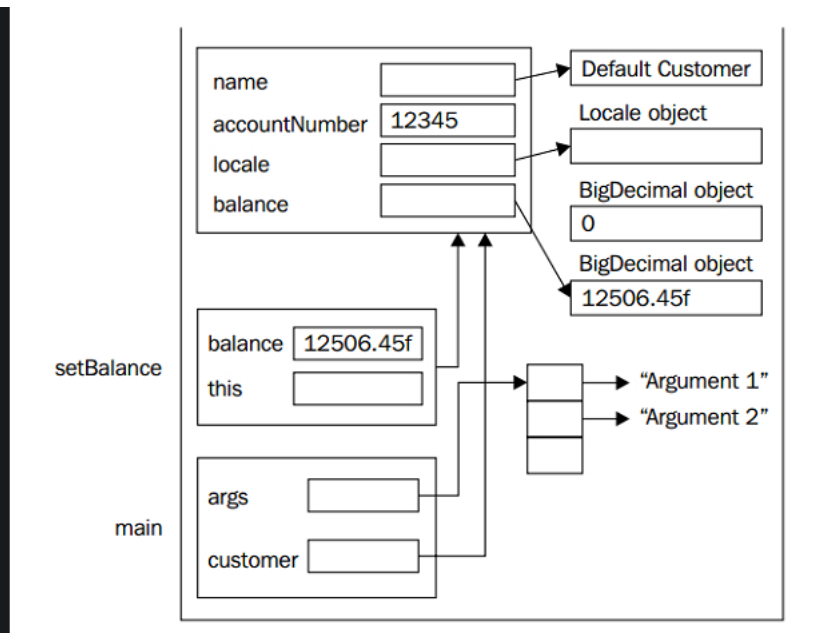
```



Khi phương thức main được thực thi, một bản ghi kích hoạt được đẩy lên bộ nhớ stack chương trình. Như được mô tả trong hình dưới, bản ghi kích hoạt của nó chỉ bao gồm tham số args duy nhất và biến tham chiếu customer. Khi thể hiện của lớp Customer được tạo ra, một đối tượng được tạo ra và phân bổ trên bộ nhớ heap. Trạng thái của stack và heap được lập lại trong ví dụ này xảy ra sau khi constructor Customer thực thi. Biến tham chiếu args trỏ tới một mảng, mỗi phần tử của mảng tham chiếu một chuỗi đại diện cho đối số khi nhập trên dòng lệnh của ứng dụng. Trong ví dụ thể hiện trong ảnh dưới đây, giả sử có hai đối số dòng lệnh là Argument 1 và Argument 2:



Khi phương thức setBalance() được thực thi, bản ghi kích hoạt của nó sẽ được đẩy lên stack chương trình như minh họa dưới đây. Phương thức setBalance() có một tham số duy nhất, balance, được gán cho biến instance balance. Nhưng trước tiên, nó được sử dụng như một đối số cho constructor BigDecimal. Từ khóa this tham chiếu đến đối tượng hiện tại.



Heap là bộ nhớ được phân bổ động cho các đối tượng. Trình quản lý heap kiểm soát cách bộ nhớ này được tổ chức. Khi một đối tượng không còn cần thiết, một thủ tục thu gom rác sẽ thực thi để giải phóng bộ nhớ để nó có thể được sử dụng lại. Trước khi một đối tượng được xử lý, phương thức `finalize` của đối tượng được thực thi. Tuy nhiên, không có gì đảm bảo rằng phương thức này sẽ được thực thi khi chương trình chấm dứt mà không cần thực hiện thủ tục thu gom rác. Đối tượng `BigDecimal` ban đầu sẽ bị phá hủy.

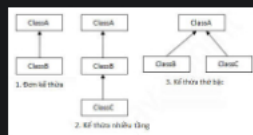
🕒 Học Java cơ bản

Immutable String trong java 🕒

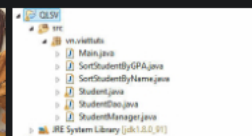


Lano Thương Hiệu Đồ Da Uy Tín

📍 Quận Đống Đa, Hà Nội

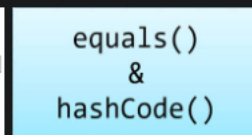


Tính kế thừa trong java - học Java cơ bản đến nâng cao



Bài tập quản lý sinh viên trong java - bài tập java có lời giải

vieltuts.vn

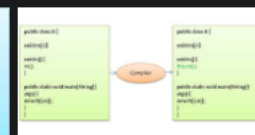


Phương thức `equals()` và `hashCode()` trong java



Biến trong java - học Java cơ bản đến nâng cao

vieltuts.vn



Từ khóa `this` trong java - học Java cơ bản đến nâng cao



Tuyển Dụng Công Nghệ Thông Tin

vieltuts.vn



Tính đa hình trong java - học Java miễn phí hay nhất



Advanced Analytics.
A Vimeo Feature

Get smart

0 bình luận

Sắp xếp theo

Cũ nhất ▾



Thêm bình luận...



Plugin bình luận trên Facebook

Danh sách bài học

[Học java](#)
[Học servlet](#)
[Học jsp](#)
[Học Hibernate](#)
[Học Struts2](#)
[Học Spring](#)
[Học SQL](#)

Câu hỏi phỏng vấn

[201 câu hỏi phỏng vấn java](#)
[25 câu hỏi phỏng vấn servlet](#)
[75 câu hỏi phỏng vấn jsp](#)
[52 câu hỏi phỏng vấn Hibernate](#)
[70 câu hỏi phỏng vấn Spring](#)
[57 câu hỏi phỏng vấn SQL](#)

About VietTuts.Vn

Hệ thống bài học trên VietTuts.Vn bao gồm các bài lý thuyết và thực hành về các công nghệ java và công nghệ web. Các bài lý thuyết trên hệ thống VietTuts.Vn được tham khảo và tổng hợp từ các trang <http://javatpoint.com>, <http://www.tutorialspoint.com>, <http://docs.oracle.com/en> ...