**Properties**

**1. Properties là gì?**

Properties là lớp phụ của lớp Hashtable sử dụng để duy trì các danh sách của các value trong đó key là một String và value cũng là một String.

Properties thường được dùng để lưu cấu hình chương trình (file config) để khi có thay đổi ta chỉ cần thay đổi trong file config và restart lại chương trình chứ không cần build lại source code và deloy lại project.

**2. Constructor**

Properties cung cấp 2 constructor:

+ Properties: tạo ra đối tượng Properties mà không có giá trị mặc định.

+ Properties(Properties proDefault): tạo đối tượng Properties mà sử dụng proDefault cho các giá trị mặc định của nó.

**3. Phương thức:**

Ngoài những phương thức được thừa kế từ HashTable thì Properties có đĩnh nghĩa thêm những phương thức sau:

|  |  |
| --- | --- |
| **STT** | **Phương thức và Miêu tả** |
| 1 | **String getProperty(String key)**  Trả về value mà liên kết với key. Một đối tượng null được trả về nếu key hoặc không có trong danh sách hoặc không có trong property list mặc định |
| 2 | **String getProperty(String key, String defaultProperty)**  Trả về value mà liên kết với key. defaultPropperty được trả về nếu key hoặc không có trong danh sách hoặc không có trong property list mặc định |
| 3 | **void list(PrintStream streamOut)**  Gửi property list tới output stream mà liên kết tới streamOut |
| 4 | **void load(InputStream streamIn) throws IOException**  Nhập một property list từ input stream mà liên kết tới streamIn |
| 5 | **Enumeration propertyNames( )**  Trả về một bản liệt kê các key. Nó cũng bao gồm các key được tìm thấy trong property list mặc định |
| 6 | **Object setProperty(String key, String value)**  Liên kết value với key. Trả về value trước mà liên kết với key, hoặc trả về null nếu không có liên kết nào tồn tại |
| 7 | **void store(OutputStream streamOut, String description)**  Sau khi ghi chuỗi được xác định bởi description, property list được ghi tới output stream mà liên kết tới streamOut |

Phương thức của hashtable:

|  |  |
| --- | --- |
| **STT** | **Phương thức và Miêu tả** |
| 1 | **void clear( )**  Reset và làm trống hash table |
| 2 | **Object clone( )**  Trả về một bản sao của đối tượng đang gọi |
| 3 | **boolean contains(Object value)**  Trả về true nếu value cân bằng với value tồn tại bên trong hash table. Trả về false nếu value đó không được tìm thấy |
| 4 | **boolean containsKey(Object key)**  Trả về true nếu key cân bằng với key tồn tại bên trong hash table. Trả về false nếu key đó không được tìm thấy |
| 5 | **boolean containsValue(Object value)**  Trả về true nếu value cân bằng với value tồn tại bên trong hash table. Trả về false nếu value đó không được tìm thấy |
| 6 | **Enumeration elements( )**  Trả về một bản liệt kê các value được chứa trong hash table |
| 7 | **Object get(Object key)**  Trả về đối tượng mà chứa value mà liên kết với key. Nếu key không có trong hash table, một đối tượng null được trả về |
| 8 | **boolean isEmpty( )**  Trả về true nếu hash table là trống; nếu nó chứa ít nhất một key thì trả về false |
| 9 | **Enumeration keys( )**  Trả về một bản liệt kê các key được chứa trong hash table |
| 10 | **Object put(Object key, Object value)**  Chèn một key và một value vào trong hash table. Trả về null nếu key đã không có trong hash table; trả về value trước mà liên kết với key nếu key đã có trong hash table |
| 11 | **void rehash( )**  Tăng kích cỡ của hash table và làm lại mới tất cả các key của nó |
| 12 | **Object remove(Object key)**  Gỡ bỏ key và value của nó. Trả về value mà liên kết với key. Nếu key không có trong hash table, một đối tượng null được trả về |
| 13 | **int size( )**  Trả về số entry trong hash table |
| 14 | **String toString( )**  Trả về biểu diễn chuỗi tương đương của một hash table |

\* hashtable phiên bản đầu là sự triển khai của Dictionary nhưng từ phiên bản java 2 đã được thiết kế lại để nó cũng có thể triển khai từ Map Interface.

\*