**Apache Commons EqualsBuilder and HashCodeBuilder ---** www.javaworld.com

public enum ExampleCollectionTypeEnum **{**

ARRAY\_LIST**,**

HASH\_SET

**}**

**===============================================**

**import** org**.**apache**.**commons**.**lang3**.**builder**.**EqualsBuilder**;**

**import** org**.**apache**.**commons**.**lang3**.**builder**.**HashCodeBuilder**;**

**import** org**.**apache**.**commons**.**lang3**.**builder**.**ToStringBuilder**;**

public class SimpleDataExample **{**

private final Long id**;**

private final String name**;**

public SimpleDataExample**(**

final Long newId**,** final String newName**)** **{**

**this.**id **=** newId**;**

**this.**name **=** newName**;**

**}**

public Long getId**()** **{**

**return** **this.**id**;**

**}**

public String getName**()** **{**

**return** **this.**name**;**

**}**

@Override

public int hashCode**()** **{**

**return** **new** HashCodeBuilder**()**

**.**append**(this.**id**)**

**.**append**(this.**name**)**

**.**toHashCode**();**

**}**

@Override

public boolean equals**(**Object obj**)** **{**

**if** **(**obj **instanceof** SimpleDataExample **==** **false)** **{**

**return** **false;**

**}**

**if** **(this** **==** obj**)** **{**

**return** **true;**

**}**

final SimpleDataExample otherObject **=** **(**SimpleDataExample**)** obj**;**

**return** **new** EqualsBuilder**()**

**.**append**(this.**id**,** otherObject**.**id**)**

**.**append**(this.**name**,** otherObject**.**name**)**

**.**isEquals**();**

**}**

@Override

public String toString**()** **{**

**return** **new** ToStringBuilder**(this)**

**.**append**(**"ID"**,** **this.**id**)**

**.**append**(**"Name"**,** **this.**name**)**

**.**toString**();**

**}**

**}**

**===============================================================**

**import** java**.**io**.**IOException**;**

**import** java**.**io**.**OutputStream**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**Collection**;**

**import** java**.**util**.**HashSet**;**

**import** org**.**apache**.**commons**.**cli**.**CommandLine**;**

**import** org**.**apache**.**commons**.**cli**.**CommandLineParser**;**

**import** org**.**apache**.**commons**.**cli**.**GnuParser**;**

**import** org**.**apache**.**commons**.**cli**.**Options**;**

**import** org**.**apache**.**commons**.**cli**.**ParseException**;**

public class App **{**

private Collection**<**SimpleDataExample**>** data**;**

private static String NEW\_LINE **=** System**.**getProperty**(**"line.separator"**);**

public static App newInstance**(**final String**[]** arguments**)** **{**

final App app **=** **new** App**();**

final ExampleCollectionTypeEnum collectionType **=** ExampleCollectionTypeEnum**.**ARRAY\_LIST**;** // parseCmdLineArgsForCollectionType(arguments);

app**.**initializeData**(**collectionType**,** System**.**out**);**

**return** app**;**

**}**

private App**()** **{**

**}**

private void initializeData**(**

final ExampleCollectionTypeEnum collectionType**,**

final OutputStream out**)** **{**

String collectionTypeStr **=** ""**;**

**switch** **(**collectionType**)** **{**

**case** ARRAY\_LIST**:**

**this.**data **=** **new** ArrayList**<**SimpleDataExample**>();**

collectionTypeStr **=** "ArrayList\_Daniel"**;**

**break;**

**case** HASH\_SET**:**

**this.**data **=** **new** HashSet**<**SimpleDataExample**>();**

collectionTypeStr **=** "HashSet"**;**

**break;**

**default:**

System**.**err**.**println**(**

collectionType **+** " is an unexpected type of collection." **+** NEW\_LINE**);**

System**.**exit**(-**1**);**

**}**

**try** **{**

out**.**write**(**

**(**"\*\*\*\*\*" **+** collectionTypeStr **+** " is being used FIRST." **+** NEW\_LINE**).**getBytes**());**

**}** **catch** **(**IOException ioEx**)** **{**

System**.**out**.**println**(**"\*\*\*\*\*" **+** collectionTypeStr **+** " is being used SECOND."**);**

**}**

**this.**data**.**add**(new** SimpleDataExample**(**1L**,** "The First One"**));**

**this.**data**.**add**(new** SimpleDataExample**(**5L**,** "The Fifth Element"**));**

**this.**data**.**add**(new** SimpleDataExample**(**3L**,** "The Third Kind of Encounter"**));**

**this.**data**.**add**(new** SimpleDataExample**(**2L**,** "The Second of a Kind"**));**

**this.**data**.**add**(new** SimpleDataExample**(**4L**,** "Fourth is a Quarter"**));**

**}**

/\*\*

\* Demonstate use of equals() method and hashCode() method implementations.

\* **@param** out OutputStream to which to write the results of the test.

\*/

public void testForCollectionContainment**(**final OutputStream out**)** **{**

final SimpleDataExample simple1 **=** **new** SimpleDataExample**(**1L**,** "The First DANIEL"**);**

final SimpleDataExample simple2 **=** **new** SimpleDataExample**(**1L**,** "The First LUCIA"**);**

**try** **{**

**for** **(**final SimpleDataExample simple **:** **this.**data**)** **{**

out**.**write**((**"COMPARISON" **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"----------" **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"First Object: " **+** simple1 **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"Second Object: " **+** simple **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"\tEqual?: " **+** simple1**.**equals**(**simple**)** **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"\t==?: " **+** **(**simple1 **==** simple**)** **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"\tHash Codes: " **+** simple1**.**hashCode**()** **+** " vs " **+** simple**.**hashCode**()).**getBytes**());**

out**.**write**(**NEW\_LINE**.**getBytes**());**

**}**

**for** **(**final SimpleDataExample simple **:** **this.**data**)** **{**

out**.**write**((**"COMPARISON" **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"----------" **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"First Object: " **+** simple2 **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"Second Object: " **+** simple **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"\tEqual?: " **+** simple2**.**equals**(**simple**)** **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"\t==?: " **+** **(**simple2 **==** simple**)** **+** NEW\_LINE**).**getBytes**());**

out**.**write**((**"\tHash Codes: " **+** simple2**.**hashCode**()** **+** " vs " **+** simple**.**hashCode**()).**getBytes**());**

out**.**write**(**NEW\_LINE**.**getBytes**());**

**}**

**}** **catch** **(**IOException ioEx**)** **{**

System**.**err**.**println**(**

"Error encountered while trying to write out two objects to be "

**+** "compared and the result of their comparison: " **+** NEW\_LINE

**+** ioEx**.**getMessage**());**

**}**

**}**

/\*\*

\* Write this application's output header/title to the provided output stream.

\*

\* **@param** out OutputStream to which to write application header information.

\*/

public static void printAppHeader**(**final OutputStream out**)** **{**

final String separatorString **=**

"==================================================================="

**+** NEW\_LINE**;**

final String headerText **=**

"= Example associated with blog entry 'Apache Commons" **+** NEW\_LINE

**+** "= EqualsBuilder and HashCodeBuilder' from Dustin's Software" **+** NEW\_LINE

**+** "= Development Cogitations and Speculations" **+** NEW\_LINE

**+** "= (http://marxsoftware.blogspot.com)." **+** NEW\_LINE**;**

**try** **{**

out**.**write**(**separatorString**.**getBytes**());**

out**.**write**(**headerText**.**getBytes**());**

out**.**write**(**separatorString**.**getBytes**());**

**}** **catch** **(**IOException ioEx**)** **{**

System**.**out**.**print**(**separatorString **+** headerText **+** separatorString**);**

**}**

**}**

/\*\*

\* Build up the command-line handling GNU-style options.

\*

\* **@return** Command-line handling options.

\*/

private static Options buildGnuOptions**()** **{**

final Options gnuOptions **=** **new** Options**();**

gnuOptions**.**addOption**(**"c"**,** **true,** "Collection Implementation Type to Use"**);**

**return** gnuOptions**;**

**}**

/\*\*

\* Apply Apache Commons CLI GnuParser to command-line arguments.

\*

\* **@param** commandLineArguments Command-line arguments to be processed with

\* Gnu-style parser.

\* **@return** Collection implementation type.

\*/

private static ExampleCollectionTypeEnum parseCmdLineArgsForCollectionType**(**

final String**[]** commandLineArguments**)** **{**

final CommandLineParser cmdLineGnuParser **=** **new** GnuParser**();**

final Options gnuOptions **=** buildGnuOptions**();**

ExampleCollectionTypeEnum collectionType **=** **null;**

String optionStr **=** ""**;**

CommandLine commandLine**;**

**try** **{**

commandLine **=** cmdLineGnuParser**.**parse**(**gnuOptions**,** commandLineArguments**);**

**if** **(**commandLine**.**hasOption**(**"c"**))** **{**

optionStr **=** commandLine**.**getOptionValue**(**"c"**);**

collectionType **=** ExampleCollectionTypeEnum**.**valueOf**(**optionStr**);**

**}**

**}** **catch** **(**ParseException parseException**)** // checked exception

**{**

System**.**err**.**println**(**

"Encountered exception while parsing using GnuParser:\n"

**+** parseException**.**getMessage**());**

**}** **catch** **(**IllegalArgumentException illegalArgEx**)** **{**

System**.**err**.**println**(**optionStr **+** " is not a valid value for -c option."**);**

final ExampleCollectionTypeEnum**[]** possibleValues **=**

ExampleCollectionTypeEnum**.**values**();**

System**.**err**.**print**(**"\tPlease use one of these values instead: "**);**

**for** **(**final ExampleCollectionTypeEnum value **:** possibleValues**)** **{**

System**.**err**.**print**(**value **+** " "**);**

**}**

System**.**err**.**println**(**"\n"**);**

System**.**exit**(-**2**);**

**}**

**return** collectionType**;**

**}**

/\*\*

\* App executable to run examples for Apache Commons EqualsBuilder and

\* HashCodeBuilder.

\*

\* **@param** arguments The command line arguments: -c with a value should be

\* specified where the value passed in with -c is the collection

\* implementation type (such as HASH\_SET or ARRAY\_LIST).

\*/

public static void main**(**final String**[]** arguments**)** **{**

printAppHeader**(**System**.**out**);**

final App me **=** App**.**newInstance**(**arguments**);**

me**.**testForCollectionContainment**(**System**.**out**);**

**}**

**}**

**........................................................................**

**===================================================================**

**=** Example associated with blog entry 'Apache Commons

**=** EqualsBuilder and HashCodeBuilder' from Dustin's Software

**=** Development Cogitations and Speculations

**=** **(**http**:**//marxsoftware.blogspot.com).

**===================================================================**

**\*\*\*\*\***ArrayList\_Daniel is being used FIRST**.**

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@3d71d552**[**ID**=**1**,**Name**=**The First DANIEL**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@1cf4f579**[**ID**=**1**,**Name**=**The First One**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 675693484 vs 702165685

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@3d71d552**[**ID**=**1**,**Name**=**The First DANIEL**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@4ee285c6**[**ID**=**5**,**Name**=**The Fifth Element**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 675693484 vs 1718548326

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@3d71d552**[**ID**=**1**,**Name**=**The First DANIEL**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@621be5d1**[**ID**=**3**,**Name**=**The Third Kind of Encounter**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 675693484 vs 1087926502

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@3d71d552**[**ID**=**1**,**Name**=**The First DANIEL**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@573fd745**[**ID**=**2**,**Name**=**The Second of a Kind**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 675693484 vs 2097031442

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@3d71d552**[**ID**=**1**,**Name**=**The First DANIEL**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@15327b79**[**ID**=**4**,**Name**=**Fourth is a Quarter**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 675693484 vs 1513975290

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@4f2410ac**[**ID**=**1**,**Name**=**The First LUCIA**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@1cf4f579**[**ID**=**1**,**Name**=**The First One**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 445434529 vs 702165685

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@4f2410ac**[**ID**=**1**,**Name**=**The First LUCIA**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@4ee285c6**[**ID**=**5**,**Name**=**The Fifth Element**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 445434529 vs 1718548326

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@4f2410ac**[**ID**=**1**,**Name**=**The First LUCIA**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@621be5d1**[**ID**=**3**,**Name**=**The Third Kind of Encounter**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 445434529 vs 1087926502

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@4f2410ac**[**ID**=**1**,**Name**=**The First LUCIA**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@573fd745**[**ID**=**2**,**Name**=**The Second of a Kind**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 445434529 vs 2097031442

COMPARISON

**----------**

First Object**:** com**.**codezone**.**SimpleDataExample@4f2410ac**[**ID**=**1**,**Name**=**The First LUCIA**]**

Second Object**:** com**.**codezone**.**SimpleDataExample@15327b79**[**ID**=**4**,**Name**=**Fourth is a Quarter**]**

Equal**?:** **false**

**==?:** **false**

Hash Codes**:** 445434529 vs 1513975290

===========================================================

**import** org**.**apache**.**commons**.**lang3**.**builder**.**HashCodeBuilder**;**

public class Person **{**

String name**;**

int age**;**

boolean smoker**;**

public Person**(**String name**,** int age**,** boolean smoker**)** **{**

**this.**name **=** name**;**

**this.**age **=** age**;**

**this.**smoker **=** smoker**;**

**}**

public String getName**()** **{**

**return** name**;**

**}**

public void setName**(**String name**)** **{**

**this.**name **=** name**;**

**}**

/\*This class enables a good hashCode method to be built for any class. It follows the rules laid out in the book Effective Java by Joshua Bloch. Writing a good hashCode method is actually quite difficult. This class aims to simplify the process.

The following is the approach taken. When appending a data field, the current total is multiplied by the multiplier then a relevant value for that data type is added. For example, if the current hashCode is 17, and the multiplier is 37, then appending the integer 45 will create a hash code of 674, namely 17 \* 37 + 45.

All relevant fields from the object should be included in the hashCode method. Derived fields may be excluded. In general, any field used in the equals method must be used in the hashCode method.

If required, the superclass hashCode() can be added using appendSuper(int).

Alternatively, there is a method that uses reflection to determine the fields to test. Because these fields are usually private, the method, reflectionHashCode, uses AccessibleObject.setAccessible to change the visibility of the fields. This will fail under a security manager, unless the appropriate permissions are set up correctly. It is also slower than testing explicitly.

A typical invocation for this method would look like:

public int hashCode() {

return HashCodeBuilder.reflectionHashCode(this); CON DIFFERENT OUTPUT!!!

}

\*/

public int hashCode**()** **{**

// you pick a hard-coded, randomly chosen, non-zero, odd number

// ideally different for each class

**return** **new** HashCodeBuilder**(**17**,** 37**).**

append**(**name**).**

append**(**age**).**

append**(**smoker**).**

toHashCode**();**

**}**

public static void main**(**String**[]** args**)** **{**

Person person **=** **new** Person**(**"Daniel"**,** 33**,** **false);**

System**.**out**.**println**(**person**.**hashCode**());** // 682968794

**}**

**}**