

```

/**
 * Interface Name      : IAddress.java
 * Version             : Build 01.10
 * Date of Change      : 2000 MAR 28
 * Author              : Don Baldwin
 * Copyright            : 2000 ASR Strategic Resources
 * Description          : An interface that sets the ground rules for the rest of
 *                      the address classes to follow.
 *                      This interface specifies the required methods
 *                      and method parameter lists for Address subclasses.
 * Version History :
 *
 * Version      Date      Author
 * Build 01.10  2000 MAR 28  Don Baldwin
 *                      Added comments
 * Build 01.00  2000 MAR 07  Anders Nyström
 *                      Added the basic method signature for all classes
 *                      that use Addresses.
 */

```

```
import java.sql.SQLException;
```

```
public interface IAddress
{
```

```

/**
 * Mutator method used to set the relational identifier of an Address
 * object. The relational identifier is the primary key of the RDBMS
 * table that is used to store Address data objects.
 * This method is a set once method. If the current value of the identifier
 * is a value greater than zero, then the value of the identifier is not
 * changed.
 * If the anID argument is zero or negative, an Illegal Argument Exception
 * will be thrown.
 * Preconditions: The current value of the identifier must be zero for
 * a valid identifier update.
 */

```

```
public void setId(long aLongID) throws IllegalArgumentException;
```

```

/**
 * Query method that returns an Address object's identifier.
 */

```

```
public long getId();
```

```

/**
 * Mutator method used to set the reference variable that points to a String
 * object used to store the Address object's first name.
 * If the argument to this function is either null or an empty string,
 * then an IllegalArgumentException is thrown.
 * Post conditions: The person has an assigned first name.
 */

```

```
public void setFirstName(String aStrFirstName) throws IllegalArgumentException;
```

```

/**
 * Query method that returns an Address object's first name.
 */

```

```
public String getFirstName();
```

```

/**
 * Mutator method used to set the reference variable that points to a String object
 * used to store the person's last name.
 * If the argument to this function is either null or an empty string, then
 * an IllegalArgumentException is thrown.
 */

```

```
public void setLastName(String aStrLastName) throws IllegalArgumentException;
```

```

/**
 * Query method that returns an Address object's first name.
 */
public String getLastName();

/**
 * Query method that returns a full person name in first name, space, and
 * last name order.
 * For example: Don Baldwin (Where Don is first name and Baldwin is last name)
 */
public String getFirstLastName();

/**
 * Query method that returns a full person name in last name seperated by a
 * comma and one space and first name.
 * For example: Baldwin, Don (Where Don is first name and Baldwin is last name)
 */
public String getLastFirstName();

/**
 * Mutator method used to set the reference variable that points to a String
 * object used to store the company's first name.
 */
public void setCompanyName(String aStrCompanyName);

/**
 * Query method that returns an Address object's company name.
 */
public String getCompanyName();

/**
 * This method returns a String object reference variable that contains the
 * Address object's street address.
 */
public String getStreet();

/**
 * This mutator returns a String object reference variable that contains the
 * Address object's country.
 */
public String getCountry();

/**
 * Query method that returns the country code String object of the Address.
 */
public String getCountryCode();

/**
 * This abstract method is a requirement that each object implementing this method
 * must be able to return a String representation of the address in a mailing label
 * format.
 */
public String getMailingLabel();

/**
 * Mutator method used to set the reference variable that points to
 * a String object used to store an Address object's postal code.
 */
public void setPostalCode(String aStrPostalCode) throws IllegalArgumentException;

/**
 * This abstract method is a requirement that each object implementing this method
 * must be able to return a String representation of the postal code.

```

```

*/
public String getPostalCode();

/**
 * Mutator method used to set the reference variable that points to
 * a String object used to store an Address object's postal region code.
 */
public void setPostalRegionCode(String aStrPostalRegionCode) throws IllegalArgumentException;

/**
 * This abstract method is a requirement that each object implementing this method
 * must be able to return a String representation of the postal region code.
 */
public String getPostalRegionCode();

/**
 * Mutator method used to set the reference variable that points to
 * a String object used to store an Address object's postal block code.
 */
public void setPostalBlockCode(String aStrPostalBlockCode) throws IllegalArgumentException;

/**
 * This abstract method is a requirement that each object implementing this method
 * must be able to return a String representation of the postal block code.
 */
public String getPostalBlockCode();

/**
 * This abstract method is a requirement that all addresses know how to add themselves
 * to the database.
 */
public void writeAddress() throws SQLException;

/**
 * This abstract method is a requirement that each object implementing this
 * method must be able to return a String representation of the address
 * object.
 */
public String toString();

} // END IAddress.java

```