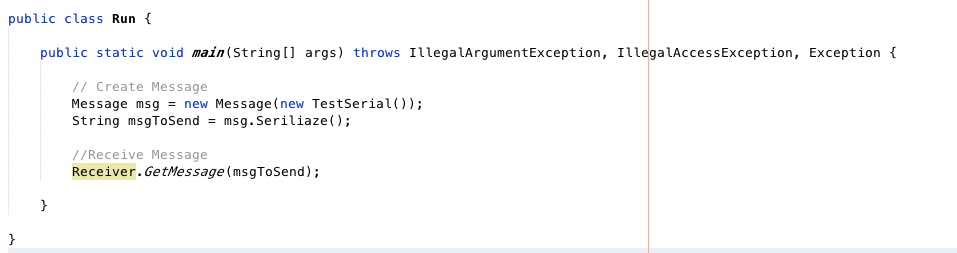
# Code Description (JAVA Part)

## Main Class (Run.Java)

* Create serialized message
* Receiver receives serialized message and finds suitable method



## Example Test Class To Be Send From Client



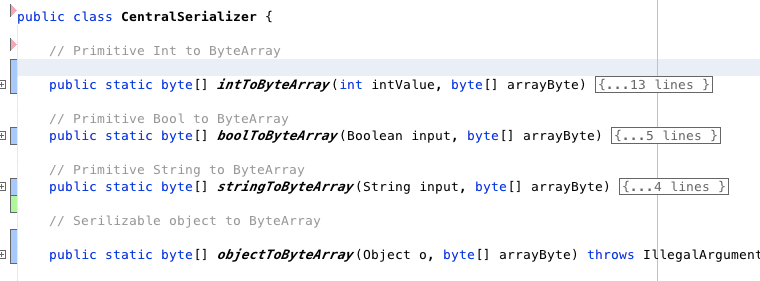
## Create a message to send Server (Message.Java)

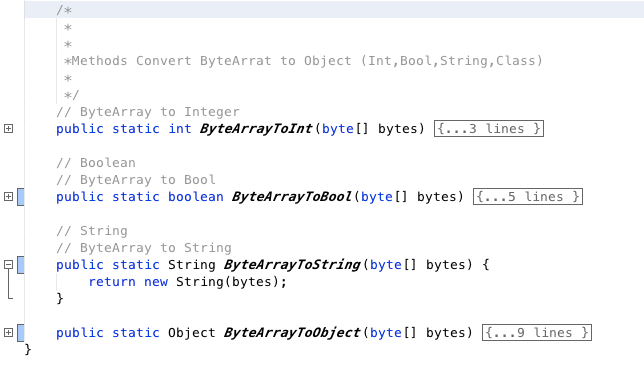
* Create Message object
* Convert object to byte array
* Serialize to string(Base64) and Deserialize to byte array



## CentralSerializer (CentralSerializer.java)

* Convert primitive or object to byte array



* Convert byte array to primitive or object(class)
* 

## ObjectSerializer (ObjectSerializer.java)

* Serialize to byte array non primitive object

Methods:

*Serialize()*

* + Creates special byte array from object using reflection
  + Byte array starts with START\_SERIALIZE (0xAC, 0xAE)
  + Then add OBJECT\_START (0x73) byte to start reading object
  + Check all fields of class and append to byte array with *CheckFields* method
  + Add OBJECT\_END (0x75) byte to end reading object
  + Finally append FINISH\_SERIALIZE (0x75) to finish serialization

## Receiver (Receiver.java)

* Receiver has available methods with *@AvaliableMethod* annotation
* Receiver has *GetMessage* method that check received byte array if it is suitable and does compliance test with using *findMethod* method.

## Annotations

*@AvaliableMethod*

* This annotation is used to recognize all methods to be checked in compliance test.

*@Mandatory*

* This annotation is used in non primitive object classes if the primitive field must appear in compliance test

## MyMethod and MyField Classes

These classes are created to help identify methods and fields in Receiver class.

## Test Classes (Test1.java, Test2.java, Test3.java)

These classes are created to test application for compliance tests.