RSAKeyPair:

```
public class RSAKeyPair {
   public RSAKeyPair(PRGen rand, int numBits)
   public RSAKey getPublicKey()
   public RSAKey getPrivateKey()
}
```

For RSAKeyPair, the bulk of the interesting work is performed by the constructor. This constuctor creates an RSA key pair. The constructor will use the PRGen called rand to get pseudorandom bits. numBits is the size in bits of each of the primes that will be used. The key pair is stored as a pair of RSAKey objects.

RSAKey:

```
public class RSAKey {
   public RSAKey(BigInteger theExponent, BigInteger theModulus)
   public BigInteger getExponent()
   public BigInteger getModulus()
   public byte[] encrypt(byte[] plaintext)
   public byte[] decrypt(byte[] ciphertext)
   public byte[] sign(byte[] message)
   public boolean verifySignature(byte[] message, byte[] signature)
   public int maxPlaintextLength()
}
```

The encrypt() method encrypts the plaintext using optimal asymmetric encryption padding (OAEP).

The decrypt() method decrypts the ciphertext.

The sign() method generates a signature (array of bytes) that can be verified by the verifySignature() method of the other RSAKey in the private/public RSAKey pair.

The verifySignature() method is used by a public RSAKey object to verify a signature generated by the corresponding private RSAKey's sign() method.

The maxPlaintextLength() method returns the largest N such that any plaintext of size N bytes can be encrypted with this key.