Lab 3: RSA encryption algorithm
This report describes the implementation of the RSA encryption algorithm.

Author
Name: Oleksii Shyshyma

Group: KH-M922в

Task
The task is to implement the RSA encryption algorithm and demonstrate its use.

Implementation
The RSA encryption algorithm is implemented in Java using the BigInteger class for arbitrary precision arithmetic. The implementation includes the generation of public and private keys, encryption and decryption of messages.

The following code shows the generation of public and private keys using random prime numbers p and q, and the commonly used value 65537 for e:

```
private final static BigInteger one = new BigInteger("1");
private final static SecureRandom random = new SecureRandom();

private BigInteger privateKey;
private BigInteger publicKey;
private BigInteger modulus;

public RSA(int bitLength) {
    BigInteger p = BigInteger.probablePrime(bitLength/2, random);
    BigInteger q = BigInteger.probablePrime(bitLength/2, random);
    BigInteger phi = (p.subtract(one)).multiply(q.subtract(one));
    modulus = p.multiply(q);
    publicKey = new BigInteger("65537"); // commonly used value for e
    privateKey = publicKey.modInverse(phi);
}
```

The encrypt and decrypt methods use the public and private keys to encrypt and decrypt messages, respectively. The encryption and decryption operations are performed using the modPow method:

```
public byte[] encrypt(byte[] message) {
    BigInteger m = new BigInteger(message);
    BigInteger c = m.modPow(publicKey, modulus);
    return c.toByteArray();
}

public byte[] decrypt(byte[] message) {
    BigInteger c = new BigInteger(message);
    BigInteger m = c.modPow(privateKey, modulus);
    return m.toByteArray();
}
```

Demo

The following code demonstrates the use of the RSA encryption algorithm by encrypting and decrypting the message "Hello, world!":

```
public static void main(String[] args) {
RSA rsa = new RSA(1024);
String message = "Hello, world!";
byte[] encrypted = rsa.encrypt(message.getBytes());
byte[] decrypted = rsa.decrypt(encrypted);
System.out.println("Original message: " + message);
System.out.println("Encrypted message: " + new String(encrypted));
System.out.println("Decrypted message: " + new String(decrypted));
}
```

The output of the demo is:

```
Original message: Hello, world!
Encrypted message:  ��vAa`Q�w/��{oC���
(Y[FD]P[FD][FD]J9#∇��T��k�����]r��G�u��|>��
Decrypted message: Hello, world!
```

The encrypted message is a byte array and is printed as a string of non-printable characters. The decrypted message is the original message "Hello, world!".