package ro**.**ase**.**ism**.**java**;** **import** java**.**io**.**FileInputStream**;** **import** java**.**io**.**FileNotFoundException**;** **import** java**.**io**.**FileOutputStream**;** **import** java**.**io**.**IOException**;** **import** java**.**security**.**AlgorithmParameters**;** **import** java**.**security**.**InvalidAlgorithmParameterException**;** **import** java**.**security**.**InvalidKeyException**;** **import** java**.**security**.**NoSuchAlgorithmException**;** **import** java**.**security**.**NoSuchProviderException**;** **import** java**.**security**.**spec**.**InvalidParameterSpecException**;** **import** java**.**util**.**Random**;** **import** javax**.**crypto**.**BadPaddingException**;** **import** javax**.**crypto**.**Cipher**;** **import** javax**.**crypto**.**IllegalBlockSizeException**;** **import** javax**.**crypto**.**NoSuchPaddingException**;** **import** javax**.**crypto**.**SecretKey**;** **import** javax**.**crypto**.**ShortBufferException**;** **import** javax**.**crypto**.**spec**.**IvParameterSpec**;** **import** javax**.**crypto**.**spec**.**SecretKeySpec**;** public class AES **{** public static String provider **=** "SunJCE"**;** public static void setProvider**(**String provider**){** AES**.**provider **=** provider**;** **}** public static void encrypt**(**String clearFile**,** String pass**,** String cipherFile**)** **throws** FileNotFoundException**,** NoSuchPaddingException**,** NoSuchProviderException**,** NoSuchAlgorithmException**,** InvalidKeyException**,** IOException**,** ShortBufferException**,** BadPaddingException**,** IllegalBlockSizeException**,** InvalidAlgorithmParameterException**{** //open input file FileInputStream fis **=** **new** FileInputStream**(**clearFile**);** //open output file FileOutputStream fos **=** **new** FileOutputStream**(**cipherFile**);** //get key bytes byte**[]** password **=** pass**.**getBytes**();** //generate a random IV //the IV size = key size = block size byte**[]** IV **=** **new** byte**[**password**.**length**];** //generate random values Random rand **=** **new** Random**();;** **for(**int i **=** 0**;**i**<**IV**.**length**;**i**++)** IV**[**i**]** **=** **(**byte**)**rand**.**nextInt**(**255**);** //write IV to file fos**.**write**(**IV**,** 0**,** IV**.**length**);** // 1 - creare cifru Cipher cipher **=** Cipher**.**getInstance**(**"AES/CBC/PKCS7Padding"**,** provider**);** // 2 - creare cheie SecretKey des64 **=new** SecretKeySpec**(**password**,** "AES"**);** // 3 - initializare cifru IvParameterSpec ivSpec**=** **new** IvParameterSpec**(**IV**);** cipher**.**init**(**Cipher**.**ENCRYPT\_MODE**,** des64**,**ivSpec**);** byte**[]** buffer **=** **new** byte**[**cipher**.**getBlockSize**()];** int noBytes **=** 0**;** byte**[]** encBlock **=** **null;** **while((**noBytes **=** fis**.**read**(**buffer**))!=-**1**){** encBlock **=** **new** byte**[**cipher**.**getOutputSize**(**noBytes**)];** int noEncBytes **=** cipher**.**update**(**buffer**,** 0**,** noBytes**,** encBlock**);** //write encrypted block fos**.**write**(**encBlock**,**0**,**noEncBytes**);** **}** encBlock **=** **new** byte**[**cipher**.**getBlockSize**()\***2 **];** //last step int noLastBytes **=** cipher**.**doFinal**(**encBlock**,**0**);** fos**.**write**(**encBlock**,**0**,**noLastBytes**);** fis**.**close**();** fos**.**close**();** **}** public static void decrypt**(**String cipherFile**,** String pass**,** String clearFile**)** **throws** FileNotFoundException**,** NoSuchPaddingException**,** NoSuchProviderException**,** NoSuchAlgorithmException**,** InvalidKeyException**,** IOException**,** ShortBufferException**,** BadPaddingException**,** IllegalBlockSizeException**,** InvalidAlgorithmParameterException**{** //open input file FileInputStream fis **=** **new** FileInputStream**(**cipherFile**);** //open output file FileOutputStream fos **=** **new** FileOutputStream**(**clearFile**);** //get key bytes byte**[]** password **=** pass**.**getBytes**();** //read the IV from file byte**[]** IV **=** **new** byte**[**password**.**length**];** fis**.**read**(**IV**,** 0**,** IV**.**length**);** // 1 - creare cifru Cipher cipher **=** Cipher**.**getInstance**(**"AES/CBC/PKCS7Padding"**,** provider**);** // 2 - creare cheie SecretKey des64 **=new** SecretKeySpec**(**password**,** "AES"**);** // 3 - initializare cifru IvParameterSpec ivSpec **=** **new** IvParameterSpec**(**IV**);** cipher**.**init**(**Cipher**.**DECRYPT\_MODE**,** des64**,**ivSpec**);** byte**[]** buffer **=** **new** byte**[**cipher**.**getBlockSize**()];** int noBytes **=** 0**;** byte**[]** encBlock **=** **null;** **while((**noBytes **=** fis**.**read**(**buffer**))!=-**1**){** encBlock **=** **new** byte**[**cipher**.**getOutputSize**(**noBytes**)];** int noEncBytes **=** cipher**.**update**(**buffer**,** 0**,** noBytes**,** encBlock**);** //write encrypted block fos**.**write**(**encBlock**,**0**,**noEncBytes**);** **}** encBlock **=** **new** byte**[**cipher**.**getBlockSize**()\***2**];** //last step int noLastBytes **=** cipher**.**doFinal**(**encBlock**,**0**);** fos**.**write**(**encBlock**,**0**,**noLastBytes**);** fis**.**close**();** fos**.**close**();** **}** **}**