Софийски Университет “Климент Охридски”
Факултет по Математика и Информатика

Контролно No. 1а

Курс: **Обектно Ориентирано Програмиране с Java**
Преподавател: д-р. Е. Кръстев **Студент** :
Дата: Декември 17, 2005 **Време за работа: 120 min**

Инструкции: Изпълнете следното задание за обектно ориентирано програмиране и предайте пълния набор от файлове необходими за решаване на програмата на флопи диск. Пълен набор от точки се присъжда за пълно решение на съответната подзадача.

Оценки:

2	**от 0 до 54 точки**
3	**от 55 до 64 точки**
4	**от 65 до 74 точки**
5	**от 75 до 84 точки**
6	**от 85 до 100 точки**

Задача 1 (100 точки)

Приложете следните принципи на Обектно ориентираното програмиране:

- ***hiding of information***
- ***software reuse***
- ***inheritance***
- ***polymorphism***

за шифроване на текстов низ от данни (***plaintext***) посредством алгоритъма на тъй наречения ***Random monoalphabetic cipher***. При този алгоритъм се използва ключов низ от текст (***cipher***) . Всяка буква в изходния ***plaintext*** се кодира като й се съпоставя буква от азбуката, чиято позиция се определя от позицията й в изходния текст и съответветствието й с буква от азбуката, която е променена от ключовия низ по следния начин.

Да предположим, че е избран низа **FEATHER** за ***cipher***. Първо се **премахват дублиращите се букви** в ***cipher*** и се получава **FEATHR**, след което се добавят **оставащите** букви от азбуката в **обратен ред.** Така се получава (***cipher pad)***

F E A T H R Z Y X W V U S Q P O N M L K J I G D C B

Тогава буквите от ***plaintext*** се шифроват като се използва **съответствието**

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

F E A T H R Z Y X W V U S Q P O N M L K J I G D C B

Например при ***plaintext:*** **WEATHER** и ***cipher:*** **FEATHER** получаваме ***ciphertext***: **GHFKYHM**

Напишете *JFrame* приложение на **Java**, която **кодира** и **декодира** даден ***plaintext*** в съответствие с по- горе **описания алгоритъм**. Всички операции за **обработка на низове да се**

сведат до работа с масиви като се **използват единствено** методите ***toCharArray***() и ***length***() на ***class String***. **Използване** на **други** ***String*** методи **не се допуска.**

Използвайте ***callback*** и ***closure*** за дефиниране на избрания мекод за шифроване като **извършете следните действия и използвате означенията за *class*, *interface*, методи, references, описани по- долу:**

a) Дефинирайте ***interface IEncryptable***, който **има два метода**

```
String encrypt(String plainText, String cipher);
// returns the encrypted string
String decrypt(String cipherText, String cipher);
// returns the decrypted string
```

Точки:3

b) Напишете ***class CipherMethod***, който **има** референция ***callbackFtn*** към ***IEncryptable*** обект и **методите**

```
String encryptText(String plainText, String cipher)
// returns the encrypted string
String decryptText(String cipherText, String cipher)
// returns the decrypted string
```

които прилагат **методите съответно за кодиране и декодиране** в зависимост от имплементацията им обекта ***callbackFtn***.

Точки:15

c) Напишете ***class Encryption*** , който имплементира ***interface IEncryptable*** във вътрешен клас (***closure***) ***class MonoEncryption***, така че ***class MonoEncryption*** *не трябва да е достъпен извън външния му* ***class Encryption***. Имплементацията на ***interface IEncryptable*** да бъде в съответствие с алгоритъма ***Random monoalphabetic cipher*** за **кодиране** и **декодиране на uppercase English text.**

За **кодиране** и **декодиране** създайте следните помощни методи в ***class MonoEncryption***

```
private int[] getCounts(String cipher){
// returns a int[26] array of counts a letter appears in cipher

}
private int[] makeCipher(String cipher){
// returns a int[26] array representing the cipher pad map
// over the standard English alphabet
// всеки елемент има стойност равна на позицията на буква
// от азбуката, която съответства на неговия индекс в cipher pad
// позицията на letter е отместването й спрямо 'A' т.е.
// letterChar - 'A'
// calls getCounts() to build the first part of the cipher pad
}
```

Точки:35

d) Добавете метод ***getMonoCipherMethod***() в ***class Encryption***, който връща обект от вътрешния клас ***MonoEncryption*** преобразувайки го нагоре до ***IEncryptable*** при *return*

Точки:5

e) Напишете ***class TestMethod,*** който е ***JFrame*** приложение. Нека това приложение **има**:

- текстови полета ***JTextField- txtKey, txtPlainText*** и ***txtCipherText.*** Полетата ***txtPlainText*** и ***txtCipherText*** които служат за въвеждане и извеждане на кодиран и декодиран текст. Полето ***txtKey*** служи за въвеждане на ***cipher*** текста
- Един ***JButton- btnEncrypt***, при натискане на който се **кодира** и отпечатва в ***txtCipherText*** въведения ***txtPlainText*** посредством метода ***encrypt()***, дефиниран в ***class MonoEncryption*** . Този метод се извиква, чрез ***callback*** в метода ***encryptText(String plainText, String cipher)*** на ***class CipherMethod (plainText*** и ***cipher*** са въведените съответно в ***txtPlainText*** и ***txtKey*** низове)
- Един ***JButton- btnDecrypt***, при натискане на който се **декодира** и отпечатва в ***txtPlainText*** въведения ***txtCipherText*** посредством метода ***decrypt()***, дефиниран в ***class MonoEncryption*** Този метод се извиква, чрез ***callback*** в метода ***decryptText(String cipherText, String cipher)*** на ***class CipherMethod (cipherText*** и ***cipher*** са въведените съответно в ***txtCipherText*** и ***txtKey*** низове***)***

Точки:32

f) **Компилирайте** и **изпълнете** приложението

Точки:10