

Sofia University  
Department of Mathematics and Informatics

Course : OO Programming with Java

Date: November 3, 2014

Student Name:

Lab No. 4

Задача 1

Напишете Java конзолно приложение за шифроване на текстов низ от данни (*plaintext*) посредством алгоритъма на тъй наречения *Caesar cipher*. При този алгоритъм всяка буква в текста за шифроване (*plaintext*) се **замества с буква отстояща на дясно от нея на *SHIFT\_LENGTH* позиции** и се получава шифрвания текст (*ciphertext*)

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

Например, в оригиналната версия на този алгоритъм *SHIFT\_LENGTH=3* и думата **TOY** (*plaintext*) се шифрова като **WRB** (*ciphertext*)- забележете, **заместването с букви в края на азбуката води до циклично продължение от началото на азбуката**

Дешифрирането се извършва по обратния път (циклично заместване на всяка буква с тази отстояща от нея на *SHIFT\_LENGTH* позиции) по зададени *ciphertext*.

Напишете конзолно приложение на Java, която **кодира** даден *plaintext* и **декодира** даден (*ciphertext*) в съответствие с по-горе описания алгоритъм.

Всички операции за обработка на низове да се сведат до **работа с масиви** като се **използват единствено** метода *toCharArray()* на *class String* за преобразуване от *String* в масив от *char*. Обратното преобразуване от *char[]* в *String* става с използване на **конструктора** *String(char[])*. **Използване** на други *String* методи **не се допуска**.

При тестването на приложението използвайте меню в текстов режим за избиране на шифроване или дешифриране

Problem No. 2.

**Write a Java application** that prompts the user to **enter a sequence of numbers** and **displays the distinct numbers** in the sequence. Assume that the input ends with 0 and 0 is not counted as a number in the sequence. (Use *class ArrayList*)

Problem No. 3.

A **simple transposition cipher** works as follows to encrypt a given string (**plaintext**). The user enters phrase and **the number of characters of the phrase** are considered as the **cipher key**. Then the user supplies the string to encrypt- the plaintext. The encryption procedure cuts the plaintext into series of characters, whose number is equal to the cipher

key, where the last portion of characters is eventually padded with spaces to the size of the cipher key. These series are stored by rows of a matrix and the encrypted text comprises the sequence of columns of the thus obtained matrix. Example of encryption:

```
plaintext:      thisistheplaintext
key text       beauty  → cipher key= 6
encryption

      thisis
      thepla
      intext
```

cipher text: ttihhnietspeilxsat

Once the user provides the key text and the cipher text the decryption reconstructs the original plain text.

Write a Java application to implement the encryption and decryption process using the `String.toCharArray()` method and arrays of `char` datatype as in the following example

```
string delimStr = " ,.: ";
char [] delimiter = delimStr.toCharArray(); // to char array
String text = new String(delimiter);        // to string
```

**DO NOT USE ANY OTHER string METHODS!**

Use **OO design** and define also a **class TransCipher** with respective data, constructors and methods, allowing you to implement the above encryption/ decryption functionality.

**Test** encryption / decryption of a string using different cipher keys.

#### Problem No. 4

За по- бързо сортиране на писма, Американската пощенска служба съветва бизнес компаниите, изпращащи големи количества от поща, да използват бар- код за означаване на ZIP кода на писмата. За целта се използва следната пет- цифрена схема за кодиране на 3- цифрен ZIP код:

Цифра/базис на кода	7	4	2	1	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	1	0	0	0	1
8	1	0	0	1	0
9	1	0	1	0	0
0	1	1	0	0	0

Така, всяка от трите цифри на ZIP - кода се представя като подреден набор от 5-те коефициента (0 или 1) пред базиса на кода според горната таблица.

Например, за представянето на цифрата 1 в 3- цифрен ZIP код използваме, че

$$1 = 0 \cdot 7 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 + 0 \cdot 1$$

и съответния ѝ бар-код е 00011 (сравнете с коефициентите за цифрата 1 в горната таблица).

Напишете `class BarCode`, което има клас данна `code`, която е двумерен масив константа и е инициализиран с 0 или 1 в съответствие с горната таблица.

Напишете метод `decode()` на `class BarCode`, който взима за аргумент трицифрено цяло число (ZIP – код), а връща `String` със съответното бар-код представяне на числото. Използвайте символа ':' за извеждане на нулите в бар кода и символа '|' за извеждане на единиците в бар кода. Така, би следвало ZIP – код 111 да се представи със следния текст

: : : | | : : : | | : : : | |

Напишете приложение, което тества `class BarCode` –

- Пита потребителя за ZIP код (трицифрено цяло число)
- Отпечатва **баркода на този** ZIP код, чрез извикване на метод `decode()` на обект от `class BarCode`

### Problem No. 5.

Computers are playing an increasing role in education. Write a program that will help an elementary school student learn multiplication. Use a `Random` object to produce two positive `onedigit` integers. The program should then prompt the user with a question, such as

How much is 6 times 7?

The student then inputs the answer. Next, the program checks the student's answer. If it is correct, display the message "Very good!" and ask another multiplication question. If the answer is wrong, display the message "No. Please try again." and let the student try the same question repeatedly until the student finally gets it right. A separate method should be used to generate each new question. This method should be called once when the application begins execution and each time the user answers the question correctly.

Use OOP design to solve the Problem.

### Problem No. 6.

Морз кодовете на английската азбука е показано на таблицата по-долу, където **точките и тиретата** представят късите и дългите сигнали по Морз.

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	--..								

Напишете `class MorseCode`, което има клас данна `code`, която е едномерен `String` масив константа и е инициализиран с **точките и тиретата** за всяка буква **A- Z**

в съответствие с горната таблица. (елементът с индекс 0 е низа “.-“, елементът с индекс 1 е низа “--.“ и т. н.)

Напишете метод `encode()` на `class MorseCode`, който взима за аргумент `String` променлива (English текст в главни букви за кодиране), а връща `String` със съответния Morse код. Разделяйте, съответните Морз кодове със един празен символ.

Напишете метод `decode()` на `class MorseCode`, който взима за аргумент `String` променлива (Morse код текст), а връща `String` със съответния English текст в главни букви. Използвайте това, че празен символ е разделител между отделните групи Морз кодове.

Напишете `Java class MorseCode Test` за тестване на всеки от тези методи.

Изведете на печат резултатите от изпълнение на тези методи, подходящо форматиращи.

Всички операции за обработка на низове да се сведат до работа с масиви като се използват единствено метода `toCharArray()` на `class String` за преобразуване от `String` в масив от `char`. Обратното преобразуване от `char[]` в `String` става с използване на конструктора `String(char[])`.

Упътване: Ако текущо прочетения символ от английския текст е `x`, то `x - 'A'` е номер на позицията в масива `code`, съдържащ Морз кода за символа `x`.

### Problem No. 7.

Дадени са един двумерен масив `table` и един едномерен масив `row` със следната структура:

```
table
3, 4, 5, 6, 3
2, 1, 2, 1, 1,
3, 2, 1, 4, 9
1, 0, 0, 1, 1
9, 2, 6, 4, 9

row
2, 1, 2, 3, 4
```

Напишете `class MathArray` с клас данни `table` и `1st`.

Инициализирайте тези клас данни със стойностите, дадени по-горе, като използване конструктор по подразбиране на `class MathArray`

- Напишете метод `multiplyMatrixByRow()` в `class MathArray`, който връща едномерния масив, получен при умножение на масива `table` по масива `row`. Използвайте вложени `for` цикли.
- Напишете метод `squareMatrix()` в `class MathArray`, който връща двумерния масив, получен при умножение на масива `table` на себе си. Използвайте вложени `for` цикли.

- Напишете метод `scalarProduct()` в `class MathArray`, който връща скаларното произведение (като число в плаваща запетая с двойно точност) на едномерния масив `row` на себе си
- Напишете метод `transposeMatrix()` в `class MathArray`, който връща транспонираната матрица на масива `table`.
- Напишете метод `matrixTrace()` в `class MathArray`, който връща сумата от елементите по диагонала на масива `table` (като число в плаваща запетая с двойно точност) на едномерния масив `row` на себе си
- Напишете метод `public String toString(int[] matrix)`, който връща `String` с елементите на едномерния масив, подходящо форматирани в редове във вида, в който например е представен масива `row` в началото на задачата
- Напишете метод `public String toString(int[][] matrix)`, който връща `String` с елементите на двумерния масив, подходящо форматирани в редове и колони във вида, в който например е представен масива `table` в началото на задачата
- Напишете метод `public String toString()`, който връща `String` със стойностите на клас данните на `table` и `lst`, подходящо форматирани в редове и колони във вида, в който са представени в началото на задачата

Напишете **Java class MathArrayTest** за тестване на всеки от тези методи.

**Изведете на печат резултатите от изпълнение на тези методи, подходящо форматирани**

### **Problem No. 8.**

Използвайте **едномерен масив** за програмиране на приложение на **Java**, което решава следната задача.

- Програмата **въвежда пет числа**, всяко от които е **между 10 и 100**, включително.
- При прочитане на **числото то се извежда на печат** (на текстовия прозорец) **само ако това число не е било въведено преди.**
- Отчетете **“най-лошия”** вариант при който **всички пет числа са различни.**
- Използвайте **масив с най-малка възможна дължина** за решаване на задачата. Извеждайте на печат пълното множество от всички различни числа, при въвеждане на ново различно от тях число.

### **Problem No. 9.**

Напишете приложение, което симулира хвърляне на две зарчета. Приложението да използва обект от **class Random** за хвърляне на първото, после и за хвърляне на второто зарче. След хвърляне на двете зарчета да се сметне сумата на точките,

показване от двете зарчета. Всяко от зарчетата показва цяло число от **1** до **6** , така че сумата от точките на двете зарчета ще се мени от **2** до **12**, при което **7** се очаква да е най- често получаваната сума, а **2** и **12** най- рядко получаваните суми.

Фигурата по- долу показва **36** възможни комбинации от сумата на точките на двете зарчетата.

Вашето приложение да симулира **36,000** хвърляния на двете зарчета. Използвайте едномерен масив за преброяване колко пъти се е паднала всяка от възможните суми на точките от двете зарчета.

**Изведете резултатите в две колонки , където първата колонка е сумата от точките, а до нея се изписва колко пъти се е паднала тази сума.**

**Добавете и каква част (в проценти) от всички хвърляния се пада на тази сума. За справка още изведете очакваното процентно съотношение за да се падне съответната сума (например, от таблицата по- долу се вижда, че има шест начина да се падне сумата **7** , така че очакваното процентно съотношение от всички хвърляния да се падне **7** е **1 / 6** )**

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

### **Problem No. 10.**

Напишете *class Sample* на **Java** и **методи** в този клас, които да реализират следното :

- Връща едномерен масив от сортираните елементи на целочислен едномерен масив, подаден като аргумент на метода като използвате метода на мехурчето
- Извеждат на печат стойността на индекса на елемент на целочислен масив, подаден като първи аргумент на метода, който съвпада със стойността на цяло число, зададено като втори аргумент на метода. При липса на съвпадение се връща -1.
- Инициализират на 8 първите 5 елемента на целочислен масив, подаден като аргумент на метода.
- Сумират първите 100 елемента на масив с плаваща запетая с двойна точност , подаден като аргумент на метода.
- Копират първите 11- елемента на целочислен масив **a** в началото на целочислен масив **b** , където масивите **a** и **b** са подадени като аргументи на метода.
- Определете най- малкия елемент на целочислен масив, подаден като аргумент на метода
- Пресметнете и върнете след приключване на изпълнението на метода скаларното произведение на два масива с плаваща запетая с двойна точност, подадени като аргументи на метода.

Напишете приложение на **Java** , което тества тези методи на *class Sample*