

## Opgave 7 : Een tekst coderen in "geheimschrift"

---

### Opgave :

Maak een Java programma dat **1 lijn tekst** kan omzetten in "geheimschrift" door volgend eenvoudig algoritme toe te passen (zie programma *VerborgenCopyright.java* op Toledo):

- neem elk teken van de originele tekst (een **String** uit een invoervak)
- als het een letter is wordt hij gecodeerd (zie verder), andere tekens blijven ongewijzigd (opm. : dit is eigenlijk niet zo verstandig ... maar is voor de eenvoud)
- stop het bekomen teken in een **StringBuffer**
- als alle tekens verwerkt zijn maak je van de **StringBuffer** weer een **String** en druk je deze af in een uitvoervak.

Het **coderen** gebeurt door bij de ASCII-code\* van elke letter een getal op de tellen , en er weer een letter van te maken (waar moet je typecasten en waar niet ?).

Dit getal noemen we de "**codeersleutel**". Komt een teken door deze optelling buiten de grenzen van het alfabet te liggen, dan wordt de waarde weer met 26 verminderd :

als de sleutel bvb. 3 is verandert een 'A' in een 'D' ( $65 + 3 = 68$ ) , maar een 'Y' verandert in een 'B' ( $89 + 3 - 26 = 66$ ). (analoog voor het bereik van de kleine letters).

Schrijf voor dit coderen een **aparte methode** met als **argumenten** de tekst + de sleutel, en als **returnwaarde** de gecodeerde tekst.

Voorzie ook een bijkomende knop voor het **decoderen** van berichtjes die je bvb. via MSN binnengekregen hebt in gecodeerde vorm (je zult ook de gebruikte sleutel moeten kennen). Maak ook hiervoor een `decodeer()` methode.

\* Tip 1 : je vindt hier een bruikbare ASCII-tabel : <http://www.asciitable.com/>

Tip 2 : je kan de boodschap nog iets "geheimer" maken door de methode **reverse()** van de klasse **StringBuffer** toe te passen.

Andere methoden uit **StringBuffer** die je allicht nodig hebt : `append()` en/of `setCharAt()`

Uit de klasse **String** heb je wellicht ook nog dit nodig :

- de lengte van een **String** bekom je met de methode **length()** (levert een *int* op)
- één bepaald teken uit een **String** bekom je met de methode **charAt( pos )** (met pos een *int* argument)

### Als dit werkt maak je volgende uitbreiding :

2. Voorzie een decodeersysteem dat gewoon alle 25 mogelijke sleutels probeert en dus 25 resultaten afdrukt, in de hoop dat de leesbare tekst er wel zal bijzitten (gebruik hiervoor een **JTextArea**). Misschien kom je dan ook te weten wat het volgende zinnetje betekent :

**NSDK-LJACBW : JGGZ CBADAWGE !** (let wel, er zit nog een extra moeilijkheidje in ;-)  
(als je de oplossing gevonden hebt hou je deze wél voor jezelf, anders is de lol ervan af !)

Dien voor het einde van de oefenzitting een verslag in via e-mail zoals in de **algemene richtlijnen** beschreven staat, ook al is je programma nog niet helemaal af.

Veel succes !

Hieronder een voorbeeld van hoe het programma er zou kunnen uitzien :

ENIGMA ... de codeerMachine

Tik hier een zin :  Sleutel :

key = 1 : Xqbbe qbbucqqb, xeu wqqj xuj tqqh ?  
key = 2 : Wpaad paatbppa, wdt vppi wti sppg ?  
key = 3 : Vozzc ozsaooz, vcs uooh vsh roof ?  
key = 4 : Unyyb nyrrznny, ubr tnng urg qnne ?  
key = 5 : Tmxxa mxxqymmx, taq smmf tqf pmmd ?  
key = 6 : Slwwz lwwpxllw, szp rlle spe olle ?  
key = 7 : Rkwy kwowkkv, ryo qkkd rod nknb ?  
key = 8 : Qjuux juunvju, qxn pjic qnc mjja ?  
key = 9 : Pittw ittmuit, pwm oiib pmb liiz ?  
key = 10 : Ohssv hsslthhs, ovl nhha ola khhy ?  
key = 11 : Ngrru grrksgr, nuk mggz nkz jggx ?  
key = 12 : Mfqqt fqqrffq, mtj lffy mly iffw ?  
key = 13 : Lepps eppeeep, lsi keex lix heev ?  
key = 14 : Kdoor doohpddo, krh jddw khw gddu ?  
key = 15 : Jcnnq cngoccn, jgg iccv jgv fct ?  
key = 16 : lbmmp bmmfnbbm, ipf hbbu ifu ebbs ?  
key = 17 : Hallo allemaal, hoe gaat het daar ?  
key = 18 : Gzkkn zkkdlzkk, gnd fzzs gds czzq ?  
key = 19 : Fyjjm yjjckyyj, fmc eyyr fcr byyp ?  
key = 20 : Exiil xiibjxi, elb dxxq ebq axxo ?  
key = 21 : Dwvhk whhaiwwh, dka cwwp dap zwwn ?  
key = 22 : Cvvgj vggzhwg, czj bwo czo ywm ?  
key = 23 : Buffi uffyguuf, biy auun byn xuul ?  
key = 24 : Ateeh teexfte, ahx zttm axm wtk ?  
key = 25 : Zsddg sddwessd, zgw yssl zwl vssj ?