|  |
| --- |
| ***Teme:***  *Izjeme:niti*  osnovno rokovanje z nitmi:  kreiranje, zagon, izvajanje procesa, kontrola zaključka |

**Naloga 1**

Program bere besedo in jo izpiše 100x. Pri tem vsako zaporedno črko besede izpisuje v lastni niti (samostojen proces). Realizirajte oba cilja:

1. realizirajte program
2. ob koncu izvajanja zadnje niti naj program izpiše: ...in konec.

**Naloga 2**

Varianta predhodnega programa: vsaka izmed črk se izpiše 10x-tolikokrat, kot je zaporedna številka črke v ASCII (8 bitni) tabeli (ASCII aka spodnjih 8 bitov unicode16). Oziroma: A se izpiše 65\*10x, C 67\*10x, ....

|  |
| --- |
| Naslednji dve nalogi lahko nadomestite tako, da za vajo 7 oddate vajo 6, z realizirano : a) štoperico, b)odštevalnikom z alarmom. Pri prvem merite čas. Pri drugem nastavite čas in sprožite alarm, ko ta čas preteče. |

**Naloga 3[[1]](#endnote-1)**

Predpostavite, da je dan zaslon velikosti 80(horizontala)x25(vertikala) znakov. Program nanj v prvi vrsti izpiše vneseno besedo, nato pa izvede padanje (animacijo) črk iz izpisane vrstice proti zadnji vrstici zaslona. Vsaka črka pada neodvisno od drugih (v lastni niti).

* Predlagam, da strukturo objekta, ki bo vseboval črko, razširite še z vertikalno in horizontalno pozicijo črke.
* Predlagam, da zaradi preglednejše vizualizacije zakasnite vsako nit po opravljenem premiku črke za kakšnih 500ms.

**Naloga 4[[2]](#endnote-2)**

Tetris med delovanje izvaja padanje likov, ter sesedanje na dnu nabrane vsebine v primeru napolnjenosti vrstic (briže zapolnjene). Naš program naj deluje podobno: začetni 'lik' je naključno zaporedje vidnih znakov iz zaloge vrednosti mehanizma base64 fiksne dolžine (do 80). Izriše se v prvi vrstici zaslona. Črke lika-'besede' se animirajo proti zadnji zapolnjeni vrstici in tam ostanejo uskladiščene. Sesedanje naj se izvede za stolpec v primeru, da je zadnja animirana črka enaka tisti, na kateri ta pristane. Vsak znak se animira neodvisno od ostalih v lastni niti.

Ko se konča animacija trenutne besede, se (z)generira nova beseda in postopek se ponovi. Generiranje nove besede se zaključi, ko višina najvišjega stolpca doseže vrstico, kjer se izpisuje začetna beseda.

Primer rezultatov izvajanja dveh faz; spodaj je stanje, zgoraj je nova beseda. Desno je rezultat predhodnega dodajanja …

123123e12 (l) 12r312310 (l)

˘ ˘

r312 0

**23123** **12** 23123312

11123eeee **1**1123e**e**ee 11123eeee

1. … [↑](#endnote-ref-1)
2. Zadnji dve nalogi silita uporabo sredstev, ki omogočojo pozicioniranje objektov na zaslon. Dve izmed (pa ne edini) poznanih možnosti sta:

   * raba JavaFX
   * raba variante knjižnice Curses (za Java predlagam: Lanterna)

   FX:

   zagotoviti 80x25 in preračun koordinat, uporaba FX specifičnih niti ( glej 05\_03\_Niti\_Procesi.zip, Multi\_nit\_09\_FX in javafx.concurrent.Task)

   Lanterna:

   Knjižnica se nahaja v Maven repozitoriju,

   v BlueJ boste namestili jar datoteko iz https://mvnrepository.com/artifact/com.googlecode.lanterna/lanterna/3.2.0-alpha1

   example 11 na <https://www.programcreek.com/java-api-examples/?api=com.googlecode.lanterna.screen.Screen> pa je skoraj vse, kar boste potrebovali [↑](#endnote-ref-2)