|  |
| --- |
| ***Teme:***  *Raba kolekcij (dinamičnih struktur)*  *Queue*  *HashMap*  *Vizualizacija (izgradnja uporabniškega vmesnika) - JavaFX* |

|  |
| --- |
| Glede na predhodno videno:  Realizacija te naloge zahteva rabo kolekcije na osnovi Queue za realizacijo 'kače' in njenega gibanja, HashMap za shranjevanje pozicij vseh ostalih aktivnih objektov igre (terminalnih elementov aaka ovir in elementov za podajševanje kače aka hrane).  Naslednja zahteva izhaja iz specifikacije tem: realizacija ni niti AWT niti Swing, zadevo implementirate v tehnologiji JavaFX.  Demonstracija rabe struktur izhaja iz demo primerov, izvedenih do vključno 9.10.2023 pri urah predmeta RSO in so pridani kot dodatek tej nalogi v datoteki z enako številko v opisu. |

**Naloga 1**

Dana demonstracija implementira kačo dolžine podano s konstrukcijsko metodo in jo po kreiranju izpiše na zaslon v obliki koordinat posameznih členov. Konstrukcija naloga podaja tudi način premikanja kače v izbrani smeri. Implementirajte mehanizme 'želvje grafike' na kači:

* naprej(int korakov)
* obrniLevo()
* obrniDesno();

slednji dve zgolj zamenjata smer, prva premakne kačo v trenutni smeri za 'korakov' korakov. Izvedena 'animacija' naj bo algoritmično krmiljena, zgolj z vizualizacijo kače po vsakem koraku. Izvedite sekvenco:

naprej(3) -> levo() -> naprej(2) -> desno() -> naprej(2)

in vizualizacijo dodajte v poročilo.

**Naloga 2**

Implementirajte vizualizacijo v JavaFx. Metode iz predhodne naloge naj bodo implementirane kot event-driven metode, prožene z gumbi aplikacije; metoda naprej/1 naj vedno izvede privzetih 5 korakov v trenutno definirani smeri. Člene kače vizualizirate s poljubnim JavaFX objektom (npr. Circle). Sama 'animacija' naj bo dosežena zgolj z dodajanjem elementov na sceno in odvzemanjem elementov s scene.

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

Pri gibanju kača lahko naleti na ovire. Za preverjanje kolizije(trka) kače z oviro potrebujemo hitro preverjanje. V ta namen uporabimo strukturo s časovnostjo iskanja O(1) – HashMap. Demo rabe je v priloženem dodatku. Hitro iskanje je zgolj v asociativnem delu; torej naj bo ključ elementa strukture pozicija.

V ravnino zgenerirajte 100 ovir in skušajte kačo navigirati preko ene izmed ovir. Vizualizirajte detekcijo morebitne kolizije.

1. če se vam da … [↑](#endnote-ref-1)