Poročilo:

O izvedeni vaji:

18\_lav03\_vaja

Laboratorijske vaje

Vaja 16

Maj Mohar, G3A

Ljubljana, marec 2025

Naloga 1

Objekt vrste Smer predstavlja trenutno smer gibanja lika, s katerim je povezan. Gibanje je ravninsko in zgolj

ortogonalno /recimo vzporedno z osmi koordinatnega sistema ravnine/.

Vektor (normaliziran) gibanja je/sta privatni lastnosti objekta in lahko zavzema samo 4 vrednosti, za vsako od smeri

po ena. Tipično objekt vrste Smer inicializiramo z smerjo gibanja, objektu, ki ga ne inicializiramo s smerjo določimo

privzeto smer gibanja 'desno'. Za vsako komponento vektorja gibanja mora obstajati vračalec/getter/. Zagotovite možnost

kontrolenega izpisa objekta na zaslon. Objekt vsebuje metodi obrniLevo/0 in obrniDesno/0, ki popravi smer tako, da v

prvem primeru 'zasuka' smer za 90 stopinj v smeri nasprotni urinemu kazalcu, in druge za 90 stopinj v smeri urinega

kazalca.

a) iz opisa konstruirajte ustrezen razred, kreirajte par objektov te vrste in preskusite:

- omejitve pri konstrukciji objekta

- zmožnost spreminjanja smeri v skladu z opisom

/ preskus in dokazovanje delovanja sta v vaši domeni /

b) izrišite razredni diagram za razred Smer

|  |
| --- |
| Smer |
| -int x  -int y |
| +Smer()  +Smer(int x, int y)  +getX()  +getY()  +obrniLevo()  +obrniDesno()  +toString() |

Koda class-a je na koncu.

Naloga 2

Lik v tej nalogi bo dokaj abstraktna zadeva. Ni opredeljen z obliko, temveč zgolj s pozicijo v ravnini in smerjo gibanja, ki jo določa vsebovan objekt vrste Smer. Seveda so vse lastnosti vsakega Lik-a privatne. Lik kreiramo tako, da mu podamo pozicijo v ravnini in smer gibanja. V primeru, da smeri ne podamo, se uporabi privzeta smer. Če Lik kreiramo brez argumentov, se postavi v izhodišče koordinatnega sistema s privzeto smerjo gibanja. Javna metoda premakni/0 premakne objekt za 1 v trenutno z objektom določeni smeri gibanja. Javni metodi obrniLevo/0 in obrniDesno/0 lik obrneta, kot je opisano v Smer. Za pozicijo lika mora/ta v objektu obstajati ustrezna vračalca.

a) iz opisa konstruirajte ustrezen razred, kreirajte par objektov te vrste in preskusite:

- premikanje

- zmožnost spreminjanja smeri pri gibanju lika

b) izrišite razredni diagram / nivo podrobnosti: omejite vsebina na zgolj javne člane

|  |
| --- |
| Lik |
| + Smer s |
| + Lik()  + Lik(int x, int y)  + Lik(int x, int y, int s1, int s2)  + getX()  + getY()  + obrniLevo()  + obrniDesno()  + premakni()  + premakni(int stevilo) |

Koda razreda Lik je na koncu.

Naloga 3

Javanski program kreira 15 objektov tipa lik in jih shrani v ustrezno tabelo. Pozicijo prvega lika ustvarite

naključno v ravnini (obseg naj bo [-50,50][-50,50]). Vsak naslednji naj bo za 3 enote oddaljen v smeri, ki jo

določa smer predhodnega.

a) Vizualizacija pozicij mora pokazati vrsto (daljica?).

b) Izvedite gibanje likov v smeri 'glave' zaporedja v tabeli (pomnite; smer je določal vedno element, ki je bil zadnji dodan v tabelo) za npr. 10 premikov.

:: nastavljalcev za objekt vrste Lik nimate, lahko pa naredite nov lik

:: če bi odstrani prvi element tabele (index 0) in dodali nov element na mesto N v tabeli, bi se zaporedje

navidezno premaknilo v smeri predhodno zadjega elementa

:: predlagam, da naredite krožno vrsto

c) preverite pravilnost delovanja še s postopkom:

- naredi 10 korakov dani smeri, nato obrniDesno, naredi še 10 korakov

A screen shot of a computer code

AI-generated content may be incorrect.

Naloga 4

Za Lik (naloga 2) in Smer (naloga 1) izdelajte minimum programske dokumentacije.

(v poročilu naj bo zgolj komentirana koda obeh razredov, ne pa tudi zgenerirana dokumentacija)

- časovnik za class v blueJ je že tak, da se iz njega avtomatično generira dokumentacija

- javadoc je standardno orodje za generiranje programske dokumentacije. Uporabite kot:

\* javadoc ime\_datoteke ali javadoc ime\_paketa

(\* javadoc Test.java ali javadoc vaja18 , pri čemer je vaja18 paket oz. direktorij s tem imenom)

orodje zgenerira dokumentacijo za brskanje na spletu (html)

\* blueJ ponuja alternativo urejevalnik/dokumentacija (izbira desno nad urejevalnikom), pri

čemer se generacija avtomatično generira ob preklopu v način dokumentacija

- dokumentator upošteva : strukturo razreda in posebno vrsto komentarjev, t.i. doc-bloke:

/\*\*

\* tole je doc block za metodo naredi

\* : metoda ne naredi nič pametnega, je tu zgolj za demonstracijo

\*

\* @param starost

\* @param velikost

\* @return novaStarost

\*

\* @author ime priimek

\* @version 0.01

\*

\* 'komentar' je oblike /\*\* --- \*/

\* med koncem doc bloka in opisom NI nobene prazne vrstice (pripadnost)

\*

\* opisujemo : razred, metodo, lastnost

\*/

int naredi(int starost, int velikost){ return novaStarost; }

**import** java.lang.\*;

**import** java.util.\*;

/\*\*

\* Opis: Class Lik shranjuje pozicijo lika in smer gibanje (abstraktno smer)

\*

\* Spremenljivke: x, y, s

\* **@author** Maj

\* **@version** 16. 3. 2025

\*/

public class Lik {

private int x;

private int y;

public Smer s;

public Lik(){

x=0;

y=0;

s = **new** Smer();

}

public Lik(int x, int y){

this.x = x;

this.y = y;

s = **new** Smer();

}

public Lik(int x, int y, int s1, int s2){

this.x = x;

this.y = y;

s = **new** Smer(s1, s2);

}

/\*\*

\* **@return** void

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Kliče funkcijo getX() iz objekta vrste Smer;

\*/

public int getX(){

**return** s.getX();

}

/\*\*

\* **@return** void

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Kliče funkcijo getY() iz objekta vrste Smer;

\*/

public int getY(){

**return** s.getY();

}

/\*\*

\* **@return** void

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Kliče funkcijo obrniLevo() iz objekta vrste Smer;

\*/

public void obrniLevo(){

s.obrniLevo();

}

/\*\*

\* **@return** void

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Kliče funkcijo obrniDesno() iz objekta vrste Smer;

\*/

public void obrniDesno(){

s.obrniDesno();

}

/\*\*

\* **@return** void

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Funkcija premakne lik v smeri gibanja, ki jo dobi iz objekta vrsta Smer

\*/

public void premakni(){

x+=s.getX();

y+=s.getY();

}

/\*\*

\* **@param** stevilo

\* **@return** void

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Funkcija premakne lik v smeri gibanja za količino stevilo, ki jo dobi iz objekta vrsta Smer

\*/

public void premakni(int stevilo){

x+=s.getX()+stevilo;

y+=s.getY()+stevilo;

}

}

**import** java.lang.\*;

**import** java.util.\*;

/\*\*

\* Opis: Class Smer shranjuje smer s pomočjo privatnih lastnosti x in y

\*

\* Spremenljivke: x, y

\* **@author** Maj

\* **@version** 16. 3. 2025

\*/

public class Smer{

private int x;

private int y;

public Smer(){

x = 1;

y = 0; //desno

}

public Smer(int x, int y){

this.x = (int)Math.signum(x);

**if**(this.x!=0)

this.y=0;

**else**

this.y = (int)Math.signum(y);

}

/\*\*

\* **@return** x

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Fukcija vrne x komponento smeri gibanja

\*/

public int getX(){

**return** this.x;

}

/\*\*

\* **@return** y

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Fukcija vrne y komponento smeri gibanja

\*/

public int getY(){

**return** this.y;

}

/\*\*

\* **@return** void

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Funkcija simulira obračanje smeri v levo s pomočjo menjave koordinat

\*/

public void obrniLevo(){

int t = y;

y=x;

x = t\*(-1);

}

/\*\*

\* **@return** void

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Funkcija simulira obračanje smeri v desno s pomočjo menjave koordinat

\*/

public void obrniDesno(){

int t =x;

x=y;

y=t\*(-1);

}

/\*\*

\* **@return** String

\*

\* **@author** Maj Mohar

\* **@version** 0.01

\*

\* Funkcija izpiše "vektor" smeri gibanja

\*/

public String toString(){

**return** ("Vector = ("+x+", "+y+")");

}

}