

SEMINARSKI RAD

**PANEVROPSKI UNIVERZITET APEIRON
FAKULTET INFORMACIONIH TEHNOLOGIJA**

**Redovne studije
Smjer „Programiranje i softversko inženjerstvo”**

**Predmet
OBJEKTNO ORIJENTISANO PROGRAMIRANJE
(SA PRIMJENOM NA PROGRAMSKOM JEZIKU JAVA)**

**„IZRADA DESKTOP IGRICE PONG
U PROGRAMSKOM JEZIKU JAVA”**

(seminarski rad)

**Predmetni nastavnik
Doc. dr Saša Salapura**

**Student
Hana Tica
Indeks br. 74-18/RITP-S**

Banja Luka, januar 2021.

Sadržaj

Uvod	3
1. Programski jezik Java	4
2. Integrirano razvojno okruženje Eclipse.....	5
3. Izrada desktop igrice Pong.....	5
3.1. Istorijski razvoj igrice Pong.....	5
3.2. Klasa PongIgrica.....	6
3.3. Klasa ProzorIgrice	7
3.4. Klasa PanellIgrice	8
3.5. Klasa Igrac	11
3.6. Klasa Lopta	13
3.7. Klasa Rezultat	14
Zaključak.....	16
Literatura.....	17

Uvod

U ovom seminarskom radu je detaljno opisan postupak izrade desktop igrice Pong u programskom jeziku Java. Početni dio rada predstavlja sažeti pregled ovog jezika i integrisanog razvojnog okruženja Eclipse, dok ostatak rada pruža uvid u kod projekta.

1. Programski jezik Java

Programski jezik Java spada u skupinu objektno – orijentisanih programskih jezika koji, kao što sam njihov naziv otkriva, fokus stavljaju na objekte. Objekti pripadaju određenoj klasi, a karakterišu ih atributi (karakteristike ili osobine) i metode (ponašanja). Drugim riječima, objektno – orijentisano programiranje podrazumijeva da se elementi programa predstavljaju kroz objekte – cjeline koje međusobno komuniciraju i sarađuju. Ovaj stil programiranja obuhvata primjenu različitih tehnika poput nasljeđivanja, enkapsulacije, polimorfizma i dr. Java je najrasprostranjeniji programski jezik ovoga tipa, a široko primjenjeni su i C#, JavaScript, PHP, Ruby i sl.

Razvoj programskog jezika Java je započeo 1991. godine, i to kao projekat namijenjen za programiranje kućanskih elektronskih uređaja. Za njegov početak su zaslužni inženjeri iz kompanije Sun Microsystems, među kojima su se istakli James Gosling i Patrick Naughton. Prvobitni naziv mu je bio Ouk (engl. Oak – hrast), te je kasnije preimenovan u Java po istomeinom brendu kafe Java Coffee. To je ujedno razlog zbog kojeg logo ovog programskog jezika prikazuje šolju tople kafe.



Slika 1. Logo programskog jezika Java

Jezik je zvanično lansiran na SunWorld konferenciji 1995. godine, a značajan je za izradu Android aplikacija, kao i različitih web aplikacija – standardnih, klijent – server, RIA (engl. Rich Internet Applications) itd.

2. Integrirano razvojno okruženje Eclipse

Desktop igrice Pong koja se analizira u ovom seminarskom radu napisana je u integrisanom razvojnom okruženju Eclipse. Osim Eclipse – a, za pisanje Java aplikacija popularan izbor predstavlja i razvojno okruženje NetBeans.

Integrirano razvojno okruženje (engl. Integrated Development Environment – IDE) Eclipse napisano je upravo u Javi, a osim za potrebe razvoja aplikacija u navedenom jeziku, koristi se i za druge programske jezike kao što su C, C++, PHP, Ruby, Python, Cobol, Ada i dr.



Slika 2. Logo integriranog razvojnog okruženja Eclipse

3. Izrada desktop igrice Pong

3.1. Istorijski razvoj igrice Pong

Pong je klasična igra koja je lansirana 29. novembra 1972. godine. Dizajnirao ju je Allan Alcorn, i to s ciljem vježbanja, a kao zadatak mu je dodijeljena od strane Nolan – a Bushnell – a, jednog od osnivača kompanije Atari. Međutim, Bushnell i drugi suvlasnik, Ted Dabney, bili su oduševljeni rezultatom Alcorn – ovog rada, te su odlučili da igricu programiraju i izdaju.

Pong je dvodimenzionalna igra koja se sastoji od dva reketa koji predstavljaju igrače i jedne lopte. Ona je ujedno sportska igra jer predstavlja simulaciju stonog tenisa. Igrači svoje rekete mogu pomjerati isključivo vertikalno, a smješteni su na krajnjoj lijevoj odnosno desnoj strani ekrana igrice. Cilj je osvojiti jedanaest bodova prije protivnika, a bod se ostvaruje kada suigrač ne uspije vratiti loptu na suprotnu stranu imaginarnog terena. Trenutni rezultat se ispisuje pri vrhu ekrana. Drugim reketom može upravljati i računar, a ne stvarni igrač.



Slika 3. Pong igrica

Radi preglednosti koda, projekat je podijeljen na sljedećih šest klasa:

- 1) klasa PongIgrica,
- 2) klasa ProzorIgrice,
- 3) klasa PanelIgrice,
- 4) klasa Igrac,
- 5) klasa Lopta i
- 6) klasa Rezultat.

3.2. Klasa PongIgrica

Klasa PongIgrica predstavlja main klasu ovog projekta jer sadrži main metodu. U slučaju ove klase, kod je veoma kratak jer ona ne sadrži atribute i dodatne metode. Main metoda sadrži samo jednu liniju koda kojom je inicijalizovan jedan objekat klase ProzorIgrice naziva prozor.

```
PongIgrica.java  ProzorIgrice.java  PanelIgrice.java  Igrac.java
1 public class PongIgrica {
2
3     public static void main(String[] args) {
4
5         ProzorIgrice prozor = new ProzorIgrice();
6
7     }
8
9 }
10
```

Slika 4. Klasa PongIgrica

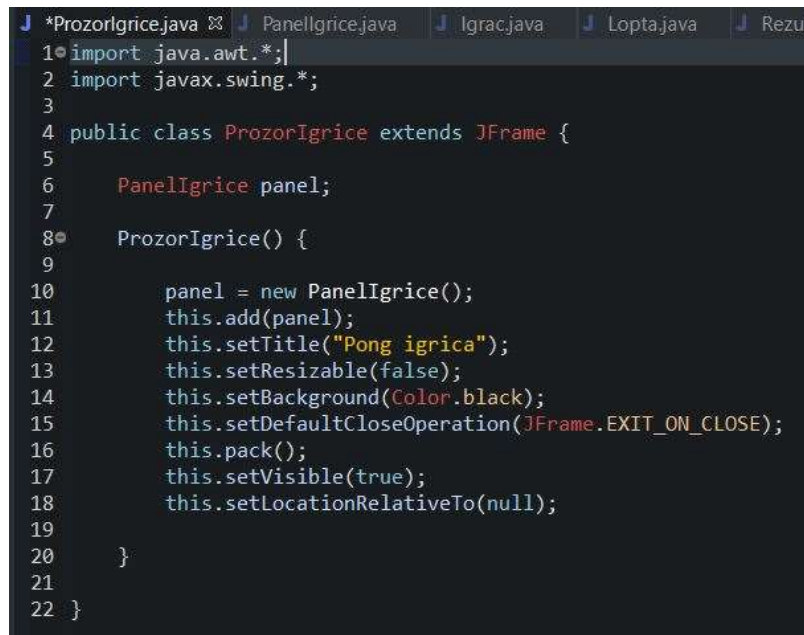
3.3. Klasa ProzorIgrice

Prvom linijom koda klase ProzorIgrice uključen je neophodni paket java.awt koji sadrži klase u vezi sa grafičkim korisničkim interfejsima (npr. klasa Button), te klasu koja omogućava crtanje po ekranu (klasa Graphics). Takođe je uključen paket javax.swing koji sadrži većinu kontrola za grafički interfejs, kao i kontrole koje korisniku omogućavaju neki izbor (npr. izbor boje).

U slučaju ove klase je primjenjena tehnika nasljeđivanja zbog čega je u okviru njenog zaglavlja napisana ključna riječ extends nakon koje je navedena klasa naziva JFrame. Na taj način je klasi ProzorIgrice omogućeno da koristi attribute i metode klase JFrame, tj. ona predstavlja podklasu svoje nadklase JFrame koja je neophodna za kreiranje prozora igrice.

Klasa ProzorIgrice, kao i prethodno opisana klasa, takođe ne sadrži attribute i dodatne metode, pored konstruktora, u okviru kojeg je inicijalizovan jedan objekat klase PanelIgrice, naziva panel (prethodno je samo deklarisan van konstruktora klase), te su pozvane sljedeće metode:

- metoda add() kojoj je kao argument proslijeđen objekat panel, te je na taj način panel "ubačen" unutar prozora igrice;
- metoda setTitle() kojom je definisan naziv prozora koji se ispisuje u gornjem lijevom uglu, a u slučaju ovog projekta taj naziv glasi "Pong igrice";
- metoda setResizable() kojoj je kao argument proslijeđena vrijednost false jer je cilj da prozor igrice uvijek bude fiksne veličine, bez mogućnosti redimenziranja;
- metoda setBackground() kojom je podešena crna boja pozadine, a koja je proslijeđena kao argument metodi putem klase Colors – Colors.black;
- metoda setDefaultCloseOperation() kojoj je kao argument proslijeđen JFrame.EXIT_ON_CLOSE što obezbjeđuje da korisnik može zatvoriti prozor igrice klikom na "X" u gornjem desnom uglu prozora;
- metoda pack() koja osigurava da prozor igrice bude odgovarajućih dimenzija, u skladu sa dimenzijama panela igrice;
- metoda setVisible() kojoj je kao argument proslijeđena vrijednost true, a koja omogućava da se prozor prikaže na ekranu;
- metoda setLocationRelativeTo() kojoj je kao argument proslijeđena vrijednost null sa ciljem da se prozor igrice uvijek pojavljuje na sredini ekrana.



```
1 import java.awt.*;
2 import javax.swing.*;
3
4 public class ProzorIgrice extends JFrame {
5
6     PanelIgrice panel;
7
8     ProzorIgrice() {
9
10         panel = new PanelIgrice();
11         this.add(panel);
12         this.setTitle("Pong igrice");
13         this.setResizable(false);
14         this.setBackground(Color.black);
15         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16         this.pack();
17         this.setVisible(true);
18         this.setLocationRelativeTo(null);
19
20     }
21
22 }
```

Slika 5. Klasa ProzorIgrice

3.4. Klasa PanelIgrice

Pored prethodno navedenih paketa, u klasu PanelIgrice su uključena još dva paketa, a to su java.awt.event koji obuhvata različite klase događaja i java.util koji obuhvata klase za podršku standardnim operacijama kolekcija, vremenskim informacijama i dr.

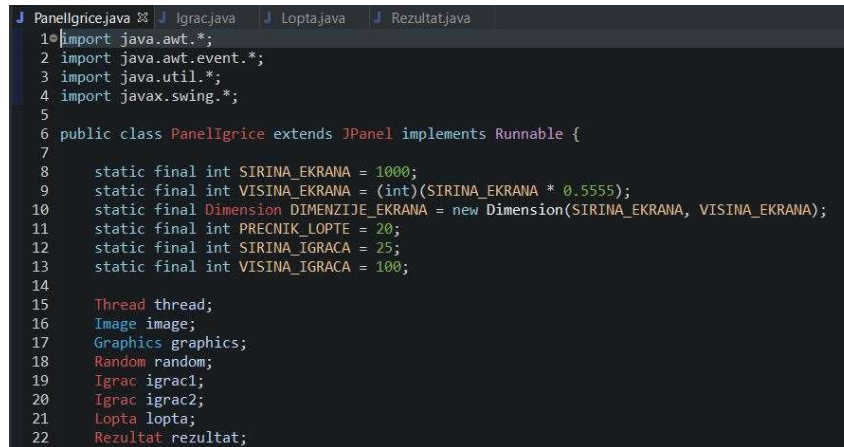
U slučaju ove klase je takođe primjenjena tehnika nasljeđivanja, tj. ova klasa predstavlja podklasu svoje nadklase JPanel koja pruža prostor unutar kojeg se mogu organizovati komponente. Takođe, u okviru zaglavlja ove klase je navedena i ključna riječ implements nakon koje je naveden naziv interfejsa Runnable. Dakle, ova klasa implementira interfejs Runnable, te definiše metodu run(), a kod koji implementira metodu run() je ono što se pokreće kao zasebna nit. Ovo predstavlja jedan od načina kreiranja niti u programu koji je napisan u jeziku Java, a drugi način bi podrazumijevao kreiranje klase koja bi zapravo bila podklasa nadklase Thread, te koja bi reimplementirala njenu metodu run().

Prije definisanja konstruktora, inicijalizovane su sljedeće globalne konstante tipa int:

- SIRINA_EKRANA i VISINA_EKRANA kojima su definisane dimenzije ekrana i pomoću kojih je inicijalizovan objekat klase Dimension,
- PRECNIK_LOPTE kojom je definisana veličina lopte,

- SIRINA_IGRACA i VISINA_IGRACA kojima su definisane širina i visina reketa koji predstavljaju igrače.

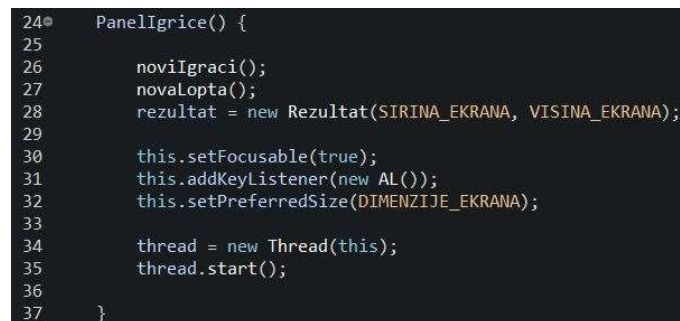
Zatim su deklarirani nužni objekti klase Thread, Image, Graphics, Random, Igrac, Lopta i Rezultat koji su kasnije inicijalizovani u okviru konstruktora, a o kojima će biti više riječi u nastavku seminarskog rada.



```
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.util.*;
4 import javax.swing.*;
5
6 public class PanelIgrice extends JPanel implements Runnable {
7
8     static final int SIRINA_EKRANA = 1000;
9     static final int VISINA_EKRANA = (int)(SIRINA_EKRANA * 0.5555);
10    static final Dimension DIMENZIJE_EKRANA = new Dimension(SIRINA_EKRANA, VISINA_EKRANA);
11    static final int PRECNIK_LOPTE = 20;
12    static final int SIRINA_IGRACA = 25;
13    static final int VISINA_IGRACA = 100;
14
15    Thread thread;
16    Image image;
17    Graphics graphics;
18    Random random;
19    Igrac igrac1;
20    Igrac igrac2;
21    Lopta lopta;
22    Rezultat rezultat;
```

Slika 6. Klasa PanelIgrice

U okviru konstruktora klase su, između ostalog, pozvane metode setFocusable() kojoj je kao argument proslijeđena vrijednost true i addKeyListener() kojom je kreiran interfejs ActionListener, a značajni su zbog reagovanja na događaje sa tastature. Takođe je pozvana metoda setPreferredSize() kojoj je kao argument proslijeđen prethodno inicijalizovani objekat DIMENZIJE_EKRANA. Svrha ove metode je, kako sam njen naziv otkriva, određivanje preferirane veličine ekrana, ali to ne znači da će ekran zasigurno biti te veličine. Pozvana je i metoda start() preko objekta thread koja pokreće izvršavanje niti.



```
24 PanelIgrice() {
25
26     noviIgraci();
27     novaLopta();
28     rezultat = new Rezultat(SIRINA_EKRANA, VISINA_EKRANA);
29
30     this.setFocusable(true);
31     this.addKeyListener(new AL());
32     this.setPreferredSize(DIMENZIJE_EKRANA);
33
34     thread = new Thread(this);
35     thread.start();
36
37 }
```

Slika 7. Konstruktor klase PanelIgrice

Klasa `PanelIgrice` sadrži sljedeće metode:

- metodu `novaLopta()` unutar koje se inicijalizuje po jedan objekat klase `Random` i `Lopta`,
- metodu `noviIgraci()` unutar koje se inicijalizuju dva objekta klase `Igrac`,
- metodu `paint()` i metodu `draw()` koje karakteriše isti parametar – objekat klase `Graphics`, a koje služe za crtanje po ekranu,
- metodu `move()` unutar koje se pozivaju istoimene metode klase `Igrac` i `Lopta`, zaslužnih za kretanje reketa i lopte,
- metodu `checkCollision()` koja obezbjeđuje da lopta i igrači prilikom kretanja ne prelaze gornju i donju granicu ekrana, da su reketa u stanju da udare loptu i tako je vrata na suprotnu stranu terena, kao i da se igraču uveća broj poena ako protivnik nije uspio odbaciti loptu,
- metodu `run()` koja sadrži petlju koja pokreće cijelu igricu.

Ova klasa takođe sadrži unutrašnju klasu koja je `ActionListener` i koja je podklasa nadklase `KeyAdapter`. Unutar ove klase se pozivaju dvije metode – `keyPressed` i `keyReleased` koje karakteriše jednak parametar, a to je objekat klase `KeyEvent`. U okviru svake od ovih 2 metoda se pozivaju istomeine metode preko odgovarajućih naziva objekata klase `Igrac`, te su zaslužne za reagovanje na događaje sa tastature na osnovu kojih se određuje kretanje reketa.

U okviru metode `checkCollision()` je definisano sljedeće:

- Ako vrijednost atributa `y` (koji predstavlja koordinatu na `y` osi) objekta `lopta` postane manji od ili jednak nuli ili veći od ili jednak `VISINA_EKRANA – PRECNIK_LOPTE`, to znači da se lopta nalazi na samom vrhu odnosno dnu ekrana, te da je potrebno preusmjeriti njeno kretanje čime se postiže efekat "odbacivanja" od vrha i dna ekrana. Isti princip je primjenjen kako bi se spriječilo "nestajanje" reketa na vrhu i dnu ekrana.
- Ako se pozivanjem metode `intersects()` utvrdi da su lopta i reket ostvarili kontakt, takođe je potrebno preusmjeriti kretanje lopte, zbog čega se poziva metoda `Math.abs()` kojoj se kao argument prosljeđuje trenutna vrijednost atributa `xBrzina` (koji predstavlja brzinu lopte na `x` osi), a koja vraća apsolutnu vrijednost tog atributa. Dakle, ako se lopta kretala ka lijevom igraču, vrijednost njene koordinate na `x` osi je postajala sve manja (`x` osa ima koordinatu 0 u gornjem lijevom uglu ekrana). Ako je ostvarila kontakt sa reketom, tj. ako je igrač uspio da je udari, vrijednost njene `x` koordinate se treba povećavati kako bi se ona počela kretati ka desnoj strani ekrana, tj. ka desnom igraču.

- Ako vrijednost atributa `x` objekta `lopta` postane manja od ili jednaka nuli, to znači da je lopta "nestala" na lijevoj strani ekrana, tj. da ju lijevi igrač nije uspio odbaciti. U tom slučaju se preko objekta klase `Rezultat` pristupa vrijednosti atributa `igrac2` (koji predstavlja broj poena desnog igrača) koji se inkrementuje tj. uvećava za 1. Sličan princip vrijedi za vođenje evidencije o ostvarenim poenima lijevoga igrača, samo što se u tom slučaju provjerava da li je vrijednost atributa `x` veća od ili jednaka `SIRINA_EKRANA – PRECNIK_LOPTE`, tj. provjerava se da li je lopta "nestala" na desnoj strani ekrana, iza desnog igrača. Potom se pozivaju metode `noviIgraci()` i `novaLopta()` jer se reketi i lopta vraćaju na svoje početne pozicije.

3.5. Klasa Igrac

Klasa `Igrac` predstavlja podklasu nadklase `Rectangle` koja specifikuje područje u koordinatnom prostoru, određeno koordinatama pravougaonika u gornjem lijevom uglu ekrana, kao i njegovom visinom i širinom.

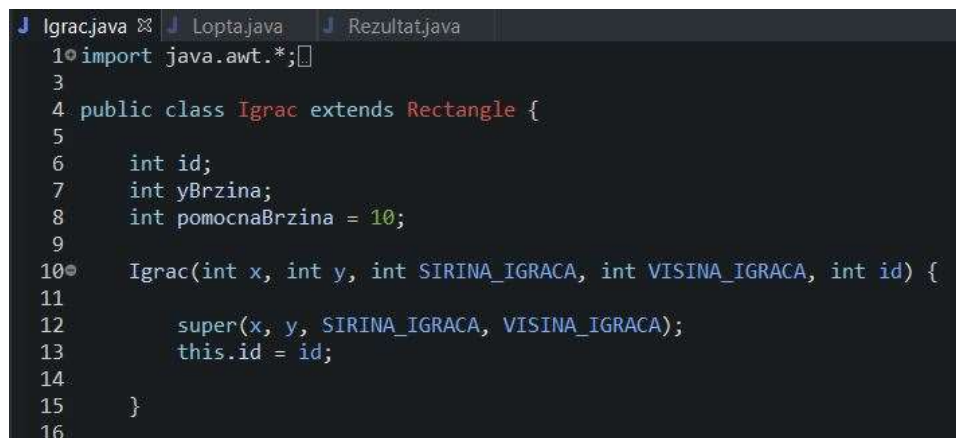
Ovu klasu karakterišu sljedeća tri atributa tipa `int`:

- atribut `id` čija je vrijednost 1 ako je riječ o lijevom reketu, a 2 ako je riječ o desnom reketu,
- atribut `yBrzina` koji predstavlja brzinu kretanja lopte po `y` osi,
- atribut `pomocnaBrzina` čija vrijednost iznosi 10 što predstavlja pomjeranje reketa za 10 piksela pri svakom pritisku odgovarajuće tipke.

Konstruktor ove klase je parametrizovan. Svi parametri su podaci tipa `int`, a to su:

- parametar `x` – vrijednost koordinate reketa na `x` osi,
- parametar `y` – vrijednost koordinate reketa na `y` osi,
- parametar `SIRINA_IGRACA` – širina reketa,
- parametar `VISINA_IGRACA` – visina reketa,
- parametar `id`.

U okviru konstruktora su ovi parametri, izuzev parametra `id`, proslijeđeni konstruktoru nadklase korištenjem ključne riječi `super`. Vrijednost parametra `id` je dodijeljena prethodno spomenutom atributu `id`.



```
1 import java.awt.*;
2
3
4 public class Igrac extends Rectangle {
5
6     int id;
7     int yBrzina;
8     int pomocnaBrzina = 10;
9
10    Igrac(int x, int y, int SIRINA_IGRACA, int VISINA_IGRACA, int id) {
11
12        super(x, y, SIRINA_IGRACA, VISINA_IGRACA);
13        this.id = id;
14    }
15 }
16
```

Slika 8. Klasa Igrac

Klasa ima sljedeće metode:

- Metodu `keyPressed()`, čiji je parametar objekat klase `KeyEvent`, unutar koje je definisana switch naredba grananja. Selektor je atribut `id` jer je bitno utvrditi o kojem se igraču radi, s obzirom na to da se svaki od njih kontroliše pomoću određenog para tipki. Zbog toga ova switch naredba ima dvije grane – case 1 i case 2. Unutar svake od njih se pozivanjem metode `getKeyCode()` provjerava da li je pritisnuta odgovarajuća tipka. Za lijevog igrača su to tipke W (prema gore) i S (prema dolje), a za desnog igrača su to tipka sa strelicom prema gore i tipka sa strelicom prema dolje. Kada je riječ o lijevom reketu, ako je igrač pritisnuo tipku W, vrijednost koordinate reketa na y osi treba da se smanjuje kako bi se on kretao prema gore. Iz tog razloga se metodi koja određuje smjer kretanja reketa prosljeđuje negativna vrijednost atributa `pomocnaBrzina` (-10). Međutim, ako je igrač pritisnuo tipku S, vrijednost koordinate reketa na y osi treba da se uvećava. U tom slučaju se metodi koja određuje smjer kretanja jednostavno prosljeđuje vrijednost atributa `pomocnaBrzina` koja iznosi 10. Jednak princip vrijedi za desnog igrača, a razlika leži u tipkama koje se koriste za usmjeravanje reketa.
- Metodu `KeyReleased()`, čiji je parametar takođe objekat klase `KeyEvent`, a koja funkcioniše na isti način. Bitno je naglasiti da je razlika u tome da, bez obzira na to koja je tipka pritisnuta, metodi koja određuje smjer kretanja reketa se uvijek kao argument prosljeđuje vrijednost nula. Razlog je taj što je cilj ove metode da zaustavi kretanje reketa.
- Metodu `setYDirection`, koja ima parametar tipa `int`, a on zapravo predstavlja vrijednost atributa `pomocnaBrzina` (pozitivnu ili negativnu vrijednost) koja se ovoj metodi

prosljeđuje u okviru dvije prethodno opisane metode. Njeno tijelo sadrži samo jednu liniju koda kojom se atributu `yBrzina` dodjeljuje vrijednost prosljeđenog argumenta.

- Metodu `move()` koja takođe sadrži samo jednu liniju koda kojom se trenutna vrijednost parametra `y` sabira sa vrijednošću atributa `yBrzina` i na taj način dobija nova vrijednost parametra `y`, tj. nova pozicija reketa na `y` osi.
- Metodu `draw()`, čiji je parametar objekat klase `Graphics`, a u okviru koje se pozivaju druge dvije metode preko naziva objekta – `setColor()` (ako je vrijednost parametra `id` 1, kao argument se prosljeđuje plava boja kako bi lijevi reket bio plave boje, a u suprotnome se kao argument prosljeđuje crvena boja) i `fillRect` kako bi se reket nacrtao na ekranu. Metodi `fillRect` se prosljeđuju četiri argumenta tipa `int`, a to su `x`, `y`, `width` (širina) i `height` (visina).

3.6. Klasa `Lopta`

Klasa `Lopta` dijeli nekoliko zajedničkih karakteristika sa prethodnom klasom, poput nasljeđivanja od iste nadklase, sličnih atributa, parametrizovanog konstruktora, metoda koje određuju smjer kretanje lopte (u okviru ove klase postoje dvije takve metode jer se lopta kreće i po `x` osi i po `y` osi), metode koja pokreće loptu, te metode za crtanje na ekran.

Značajna razlika je očigledna kada je riječ o konstruktoru ove klase. Naime, svaki put kada neki od igrača ostvari poen, pored toga što se reketi vraćaju u početnu poziciju, cilj je i da se lopta počne kretati u novom, nepredviđenom smjeru, počevši od sredine ekrana. Zbog toga je najprije bilo potrebno inicijalizovati jedan objekat klase `Random`. Takođe su inicijalizovane dvije promjenljive tipa `int` – `randomXDirection` i `randomYDirection`, a koje poprimaju vrijednosti koju vraća pozvana metoda `nextInt()` preko objekta klase `Random`. Toj metodi je kao argument prosljeđena vrijednost dva, pa ona kao vrijednost može da vrati nulu ili jedan. Ako vrati nulu, lopta će se kretati ulijevo po `x` osi, te je potrebno dekrementovati vrijednost promjenljive `randomXDirection` kako bi se to ostvarilo. U suprotnome, tj. ako vrati jedan, lopta će se u svakom slučaju kretati udesno po `x` osi. Na kraju se vrijednost promjenljive `randomXDirection` množi sa vrijednošću atributa `pomocnaBrzina` (koji u slučaju klase `Lopta` iznosi 4), a dobijeni izraz se prosljeđuje metodi koja određuje smjer kretanja lopte po `x` osi. Identičan princip se primjenjuje i za promjenljivu `randomYDirection`, odnosno za kretanje lopte po `y` osi.

Još jedna razlika jeste da se u okviru metode draw() ne poziva metoda fillRect(), već metoda fillOval(), s obzirom na to da reketi i lopta nisu jednakog geometrijskog oblika, a isti parametri se prosljeđuju kao argumenti.

```
12  Lopta(int x, int y, int width, int height) {  
13  
14      super(x, y, width, height);  
15      random = new Random();  
16      int randomXDirection = random.nextInt(2);  
17      if(randomXDirection == 0)  
18          randomXDirection--;  
19      setXDirection(randomXDirection * pomocnaBrzina);  
20      int randomYDirection = random.nextInt(2);  
21      if(randomYDirection == 0)  
22          randomYDirection--;  
23      setYDirection(randomYDirection * pomocnaBrzina);  
24  
25  }  
26
```

Slika 9. Klasa Lopta

3.7. Klasa Rezultat

Roditeljska klasa klase Rezultat je takođe Rectangle.

Karakterišu je četiri atributa tipa int, od kojih su dva globalna – atributi sirinaEkрана i visinaEkрана. Vrijednost im se dodjeljuje u okviru konstruktora klase koji kao parametre prihvata upravo navedene vrijednosti. Preostala dva atributa naziva igrač1 i igrač2 sadrže broj poena koje je ostvario prvi odnosno drugi igrač.

Ova klasa ima još samo jednu metodu, a to je metoda draw(). Pored prethodno opisane metode setColor(), u okviru ove metode su pozvane:

- metoda setFont() kojom se određuje izgled fonta za ispis rezultata;
- metoda drawLine() kojom se crta vertikalna linija na sredini ekrana, a kao argumenti joj se prosljeđuju dva para koordinata; linija se crta između dvije tačke koje se dobijaju na osnovu njih;
- metoda drawString() koja ispisuje rezultat pri vrhu ekrana; u slučaju jednocifrenog rezultata, desetica ima vrijednost nula, pa je prvo bilo potrebno proslijediti vrijednost koja se dobije dijeljenjem jednocifrenog rezultata brojem 10 (a to je 0), a zatim vrijednost koja se dobije kao ostatak pri takvom dijeljenju (jednocifreni rezultat). Takođe je bilo potrebno precizirati, i to korištenjem vrijednosti atributa sirinaEkрана i piksela, tačnu poziciju na ekranu na kojoj će se rezultat pojavljivati.

```
J *Rezultat.java 8
1 import java.awt.*;
2
3 public class Rezultat extends Rectangle {
4
5     static int sirinaEkrana;
6     static int visinaEkrana;
7     int igrac1;
8     int igrac2;
9
10 Rezultat(int sirinaEkrana, int visinaEkrana){
11     Rezultat.sirinaEkrana = sirinaEkrana;
12     Rezultat.visinaEkrana = visinaEkrana;
13 }
14
15 public void draw(Graphics g) {
16
17     g.setColor(Color.white);
18     g.setFont(new Font("Consolas",Font.PLAIN,60));
19
20     g.drawLine(sirinaEkrana / 2, 0, sirinaEkrana / 2, visinaEkrana);
21
22     g.drawString(String.valueOf(igrac1 / 10) + String.valueOf(igrac1 % 10), (sirinaEkrana / 2) -85, 50);
23     g.drawString(String.valueOf(igrac2 / 10) + String.valueOf(igrac2 % 10), (sirinaEkrana / 2) +20, 50);
24 }
25 }
```

Slika 10. Klasa Rezultat

Zaključak

Objektno – orijentisani programski jezik Java je široko zastupljen u izradi Android i web aplikacija. U ovom seminarskom radu je prikazana njegova primjena u izradi desktop igrice Pong, jedne od prvih klasičnih igrica koja je ostvarila snažan uticaj na razvoj industrije igara.

Literatura

1. <https://docs.oracle.com/javase/tutorial/>
2. <https://www.eclipse.org/ide/>
3. <https://www.britannica.com/topic/Pong>