

## Dokumentacija implementacije

Rješenje se sastoji od sljedećih klasa:

1. **Result**
2. **Direction**
3. **Ball**
4. **Paddle**

U programu su prisutni i enumi:

1. **Dir** – potreban da bi se označio smjer kretanja loptice. Moguća kretanja su: desno, lijevo, gore, dolje, gore-desno, gore-lijevo, dolje-desno, dolje-lijevo.
2. **Level** – potreban da bi se označio level kojeg je igrač odabrao.

### • Pregled metoda klasa

#### 1. Result

Atributi:

- int player = 0 – broj poena igrača
- int computer = 0 – broj poena computera

	<i>Result()</i> - konstruktor, kreira objekat rezultat i nacrtava ga prvi put
void	<i>setUp()</i> - refreshuje rezultat na vrhu ekrana
void	<i>playerScores()</i> - igrač postigao poen
void	<i>computerScores()</i> - computer postigao poen
int	<i>getPlayerScore()</i> - vraća poene igrača
int	<i>getComputerScore()</i> - vraća poene computera

## 2. Direction

Atributi:

- int x = 0 – smjer kretanja po x osi
- int y = 0 – smjer kretanja po y osi

void	<i>setAxis(Dir direction)</i> - privatna metoda , postavlja smjer obje ose za dati Dir (enum)
	<i>Direction(Dir direction)</i> - konstruktor, kreira objekat direction i postavlja ose
Direction	<i>getDirection(Dir direction)</i> - mijenja smjer na osama i vraća promijenjen
int	<i>getX()</i> - vraća smjer za x osu
int	<i>getY()</i> - vraća smjer za y osu

## 3. Ball

Atributi:

- int x – pozicija lopte po x-osi
- int y – pozicija lopte po y-osi
- int r = 5 – dijametar lopte (ostavljeno radi mogućnosti nadograđivanja)
- uint32\_t color = LCD\_COLOR\_YELLOW – boja lopte
- Level lev – izabrani nivo u igri, utiče na brzinu kretanja

	<i>Ball(uint32_t newColor, Level setLevel)</i> - konstruktor, kreira loptu i crta je prvi put
void	<i>resetBall()</i> - vraća loptu na početnu poziciju
int	<i>getX()</i> - vraća poziciju na x-osi
int	<i>getY()</i> - vraća poziciju na y-osi
void	<i>moveBall(Direction dir)</i> - pomjera loptu u zadanom smjeru

## 4. Paddle

Atributi:

- int x – pozicija reketa na x osi
- int y – pozicija reketa na y osi
- int width = 5 – sirina reketa (ostavljeno radi mogućnosti nadograđivanja)
- int height = 45 – visina reketa (ostavljeno radi mogućnosti nadograđivanja)
- uint32\_t color = LCD\_COLOR\_RED – boja reketa

	<i>Paddle(int x, int y, uint32_t color)</i> - konstruktor, kreira reket i crta ga prvi put
void	<i>resetPaddle(int setX, int setY)</i> - vraća reket na početnu poziciju
void	<i>movePaddle(int moveY)</i> - pomjera reket za zadanu visinu
int	<i>getX()</i> - vraća poziciju po x-osi
int	<i>getY()</i> - vraća poziciju po y-osi

- **Funkcije koje omogućavaju realizaciju korisničkog interfejsa**

void	<i>frontPage()</i> - crta glavnu tj početnu stranicu
void	<i>displayStay()</i> - dodatni display (lcd) popuni
void	<i>paddleOptions()</i> - crta boje za reket pri izboru
void	<i>ballOptions()</i> - crta boje za loptu pri izboru
void	<i>chooseAColorPage()</i> - crta stranicu za izbor boja
void	<i>chooseALevelPage()</i> - crta stranicu za izbor težine
void	<i>setNet()</i> - crta mrežu na sredini terena
void	<i>gameScreen ()</i> - sprema ekran za početak partije
void	<i>winPage()</i> - crta stranicu u slučaju pobjede
void	<i>gameOverPage()</i> - crta stranicu u slučaju poraza
void	<i>selectedPaddle(int x, int y)</i> - crta okvir pri izboru boje reketa
void	<i>selectedBall(int x, int y)</i> - crta okvir pri izboru boje lopte

- **Funkcije koje predstavljaju reakciju na asinhroni događaj (pritisak na taster)**

void	<i>moveDown()</i> - pomjera reket igrača prema dole
void	<i>moveUp()</i> - pomjera reket igrača prema gore

U mainu imamo više while petlji, za svaki prikaz posebne stranice po jednu, gdje provjeravamo šta je korisnik izabrao pri pritisku na ekran. Komentarima u kodu je naznačen svaki dio odvojeno i ključne metode i pozivi i dijelovi kodova su također popraćeni komentarima kako bi se razumjela logika koda.

Sistem je veoma lahko nadograditi sa silnim opcijama i modificirati te mislimo da je ispoštovano mnogo principa pisanja čitljivog i funkcionalnog koda.