

Dokumentacja projektu na przedmiot Programowanie Obiektowe

Magdalena Jarecka

rok I informatyka 308730

Implementacja gry w Ruletkę w języku Java

Spis treści

1	Wstęp	2
2	Jak posługiwać się programem?	2
3	Klasa App	2
3.1	Opis	2
4	Klasa BasicWindow	2
4.1	Opis	2
4.2	Metoda show()	3
5	Klasa MainWindow	3
5.1	Opis	3
5.2	Podklasa GameListener	3
5.3	Podklasa ExitListener	3
6	Klasa GameWindow	3
6.1	Opis	3
6.2	Metoda SetWagered()	4
6.3	Podklasa CheckBalance	4
6.4	Podklasa BackListener	4
6.5	Podklasa SpinListener	4
7	Klasa RouletteLayout	4
7.1	Opis	4
7.2	Podklasa Listener	4
7.3	Metoda add(int i, int m)	5
7.4	Metoda zeruj()	5

7.5	Podklasa FListener	5
7.6	Metoda check(int num, String c)	5
8	Klasa Wheel	5
8.1	Opis	5
9	Klasa Error	6
9.1	Opis	6

1 Wstęp

Program służy do jednoosobowej gry w Ruletkę, zachowując przy tym wszystkie jej zasady.

2 Jak posługiwać się programem?

Program należy uruchomić komendą "java App". Gracz zaczyna ze stawką 1000. Aby obstawić zakład, należy wybrać jeden z 10 przycisków znajdujących się po prawej stronie, lub wpisać w okno oznaczone jako "Number 1-36" jedną z liczb z przedziału 1-36. Następnie należy wpisać dowolną stawkę, nie przekraczającą posiadanej, w okno oznaczone jako "Bet" i wcisnąć enter. Obstawiona stawka pojawi się w dolnej tabeli. Gracz może obstawiać dowolnie, pod warunkiem, że nie obstawi więcej niż 5 numerów. Następnie należy wcisnąć przycisk Spin. Kulka wyląduje na danym polu, a z naszej posiadanej stawki zostanie adekwatnie do wyniku odjęte, lub dodane.

3 Klasa App

3.1 Opis

Klasa zawiera metodę main i służy do uruchamiania programu.

4 Klasa BasicWindow

4.1 Opis

Klasa jest klasą bazową, stworzoną na potrzeby stworzenia klas MainWindow i GameWindow. Zawiera elementy mające poprawić estetykę okna oraz metodę show(). Jej konstruktor ustawia jedynie layout oraz ustawia zamknięcie okna jako domyślną operację zakończenia programu.

4.2 Metoda show()

Funkcją tej metody jest ustawienie lokalizacji okna tak, aby pojawiło się ono na środku ekranu oraz wywołanie operacji setVisible(true).

5 Klasa MainWindow

5.1 Opis

Klasa jest pochodną klasy BasicWindow. Tworzy okno główne, w którym znajdują się dwa przyciski "Play Roulette" i "Exit". Pierwszy przenosi nas do okna gry, zaś drugi zamyka program. Jej konstruktor ustawia rozmieszczenie elementów oraz podłącza słuchaczy do przycisków.

5.2 Podklasa GameListener

Podklasa jest słuchaczem przycisku "Play Roulette". Implementuje ona interfejs ActionListener i rozszerza klasę abstrakcyjną Thread. Zawiera trzy metody:

1. Metoda actionPerformed(ActionEvent e) "chowa" okno główne, tworzy nowe okno z grą oraz wywołuje metodę start().
2. Metoda start() tworzy nowy wątek, którego zadaniem jest pilnować aby w momencie przyciśnięcia przycisku "Back" w oknie gry, wrócić do okna głównego.
3. Metoda run() jest implementuje instrukcje dla wątku stworzonego w metodzie start().

5.3 Podklasa ExitListener

Podklasa jest słuchaczem przycisku "Exit" i zamyka program jeśli zostanie on wciśnięty.

6 Klasa GameWindow

6.1 Opis

Klasa jest pochodną klasy BasicWindow. Tworzy okno, w którym znajdują się elementy potrzebne do gry. Jej konstruktor ustawia kolejno kulkę, koło rulety, przycisk Spin, etykietę z aktualnie posiadaną stawką, stół do ruletki (będący oddzielnym obiektem klasy RouletteLayout), tabelę z obstawianymi polami oraz przycisk "Back". Posiada metodę SetWagered() i trzy podklasy.

6.2 Metoda SetWagered()

Metoda służy do aktualizacji tabeli z obstawianymi polami. Pobiera ona dane ze stołu i dzięki temu na bieżąco pokazuje graczowi co i na ile obstawił.

6.3 Podklasa CheckBalance

Klasa ta rozszerza klasę abstrakcją Thread. Służy ona do bieżącego obserwowania obstawiania. Zawiera dwie metody:

1. Metoda start() tworzy nowy wątek, którego zadaniem pilnowanie aby etykita z posiadaną stawką oraz tabela z obstawionymi polami były aktualne. Wywołuje ona metodę SetWagered().
2. Metoda run() implementuje instrukcje dla threada stworzonego w metodzie start().

6.4 Podklasa BackListener

Klasa ta implementuje interfejs ActionListener i słucha przycisku "Back". W razie wciśnięcia, ustawia bool goback na true (z czego skorzysta Podklasa GameListener klasy MainWindow) i pozbywa się okna.

6.5 Podklasa SpinListener

Klasa ta implementuje interfejs ActionListener i słucha przycisku "Spin". W razie wciśnięcia wywołuje ona metodę spin() koła, która losuje numer. Następnie przekazuje numer oraz kolor do metody check() stołu ruletki i ustawia kulkę na właściwym miejscu.

7 Klasa RouletteLayout

7.1 Opis

Klasa ta rozszerza klasę JPanel. Służy do tworzenia stołu do ruletki. Znajdują się na niej przyciski służące do obstawiania, okienko tekstowe do wpisania stawki oraz okienko do wpisania numeru, jeśli taki chcemy obstawić. Posiada trzy metody i dwie podklasy.

7.2 Podklasa Listener

Jest to klasa implementująca interfejs ActionListener i służy do słuchania wszystkich przycisków znajdujących się na stole oraz okna tekstowego "Bet". W przypadku wciśnięcia uaktualnia zmienną number, która służy do tymczasowego przechowania klikniętego zakładu. Jeśli po tym gracz wpisze stawkę do okna "Bet", wywoła ona metodę add(int i, int m). Jeśli gracz

będzie próbował podać stawkę bez ówczesniejszego obstawienia, poda stawkę większą niż jest wstanie, lub nie wpisze liczby do okna "Bet" pojawi się adekwatny error.

7.3 Metoda `add(int i, int m)`

Metoda ta służy do zapisania, który zakład został obstawiony i odjęcia od salda obstawionej stawki. Wywołuje też metodę `zeruj()`, która umożliwia dalsze obstawianie. Zmienna `i` jest indeksem zakładu a `m` stawką na niego postawioną.

7.4 Metoda `zeruj()`

Ta pomocnicza metoda służy do wyczyszczenia okien tekstowych "Bet" i "Number 1-36", oraz zresetowania zmiennych `number` i `money`, które oznaczają kolejno aktualnie obstawiony zakład, oraz stawkę na niego postawioną.

7.5 Podklasa `FListener`

Jest to klasa implementująca interfejs `FocusListener` i służy do słuchania okienka "Number 1-36" Zawiera dwie metody:

1. Metoda `focusGained()` jest jedynie nadmieniona z konieczności implementacji, nie używamy jej w rzeczywistości
2. Metoda `focusLost()` określa co dzieje się podczas wpisania do czegoś do okna "Number 1-36" i kliknięcia na coś innego. Jeśli to co wpisujemy jest numerem od 1 do 36, to metoda ustawi zmienną `number` na odpowiadający jej obstawiany numer. W przypadku gdy numer nie jest z zakresu, lub gdy zostały wpisane znaki inne niż cyfry pojawi się adekwatny error.

7.6 Metoda `check(int num, String c)`

Metoda ta służy do uaktualnienia salda gracza w zależności tego co obstawił i jaki numer został wylosowany na kole. W przypadku gdy metoda zostanie wywołana, a nic nie było obstawione pojawi się error.

8 Klasa `Wheel`

8.1 Opis

Klasa `Wheel` służy do tworzenia koła rulety. Jej konstruktor ładuje do `Componentów` `Image` oraz `Ball` odpowiednio pliki `wheel.png` i `ball.png`. Posiada ona trzy metody:

1. Metoda `spin()` losuje numer oraz przydziela mu kolor na podstawie danych z tablicy `N`.
2. Metoda `x()` daje koordynat `x` kulki. Robi to odczytując dane z tablicy `N`. W tablicy `X` pod indeksem `i` kryje się położenie `i`-tego numeru na kole.
3. Metoda `y()` daje koordynat `y` kulki.

9 Klasa `Error`

9.1 Opis

Klasa ta służy do tworzenia okienek błędów, wraz z pojawieniem się jednego z możliwych błędów. Jej konstruktor tworzy okienko `Error`, bez żadnej wiadomości oraz ustawia na niej przycisk `OK`. Posiada jedną podklasę słuchającą `OK` i w razie przyciśnięcia, pozbywa się okna. Jej metody ustawiają wiadomość w okienku na różne błędy. Jest sześć możliwych wiadomości:

- `BadInput()` ustawia wiadomość "Bad input!" w przypadku gdy w okna tekstowe stołu do ruletki wpisze się coś innego niż liczby dodatnie.
- `InvalidRange()` ustawia wiadomość "Invalid Range!" w przypadku gdy w okno tekstowe "Number 1-36" stołu wpisze się coś spoza zakresu.
- `SelectBet()` ustawia wiadomość "First select bet!" w przypadku gdy gracz zechce obstawić sumę bez wcześniejszego zaznaczenia zakładu.
- `FiveNumbers()` ustawia wiadomość "Selected more than 5 numbers!" w przypadku gdy gracz próbuje obstawić więcej niż 5 numerów.
- `NothingWagered()` ustawia wiadomość "Nothing wagered!" w przypadku gdy gracz spróbuje zakreślić kołem bez obstawienia czegokolwiek.
- `TooMuch()` ustawia wiadomość "You don't have that much!" w przypadku gdy gracz spróbuje obstawić więcej niż posiada.

Wszystkie te metody wywołują też metodę `show()`.