

Błądzenie przypadkowe na płaszczyźnie

Utwórz klasę Walkerd2, która dziedziczy po klasie Particle:

```
class Particle{
private int x,y; // pozycja czastki
public Particle(int x0, int y0){ // konstruktor
x=x0;
y=y0;
}
public int get_x(){return x;} // metoda get_x
public int get_y(){return y;} // metoda get_y
public void set_x(int x0){x=x0;} // metoda set_x
public void set_y(int y0){y=y0;} // metoda set_y
} // koniec Particle
```

Klasa Walkerd2 implementuje metodę step, która przesuwa cząstkę o jeden w losowo wybranym kierunku

```
class Walkerd2 extends Particle{
    public void step(){
// TODO
    }
} // koniec Walkerd2
```

Następnie utwórz klasę Applet walk:

```
import java.awt.*;
import java.applet.*;
public class Applet_walk extends Applet{

    int applet_width=500;
    int applet_height=500;
    public void init(){
        setSize(applet_width,applet_height);
        setBackground(Color.orange);
    }
    public void paint(Graphics g){
// TODO
    }

} // koniec Applet_walk
```

W metodzie paint tworzony jest obiekt klasy Walkerd2. Pozycja początkowa - środek apletu. Następnie cząstka ta błądzi zadaną ilość kroków (n_step=1000), a jej pozycja jest odrysowywana (kolorem niebieskim) w trakcie błądzenia.

Rozwiązanie:

```
class Walkerd2 extends Particle{
    public Walkerd2(int x0, int y0){ // konstruktor klasy Walkerd2
        super(x0,y0); //konstruktor klasy bazowej:
// wymagany (jako pierwsza linijka konstruktora) jeżeli w klasie bazowej nie ma
// konstruktora zero-argumentowego
```

```

    }
    public void step(){ // metoda step
        double rand=Math.random();
        if (rand<0.25) {set_x(get_x()-1);} else
        if (rand<0.5) {set_x(get_x()+1);} else
        if (rand<0.75) {set_y(get_y()-1);} else
        {set_y(get_y()+1);}
    }
} // koniec Walkerd2

import java.awt.*;
import java.applet.*;
public class Applet_walk extends Applet{
    int applet_width=500;
    int applet_height=500;
    public void init(){
        setSize(applet_width,applet_height);
        setBackground(Color.orange);
    }
    public void paint(Graphics g){
        int i,n_step;
        n_step=1000;
        Walkerd2 w1 = new Walkerd2(applet_width/3,applet_height/3);
        g.drawString("Błądzenie przypadkowe (d=2)",60,25);
        g.setColor(new Color (0,0,255));
        for (i=1;i<=n_step;i++){
            w1.step();
            g.drawRect(w1.get_x(),w1.get_y(),1,1);
        }
    }
} // koniec Applet_walk

```

Dodaj jeszcze jedną błądzącą cząstkę, której położenie odrysowywane jest innym kolorem. Niech kolor cząstki będzie jedną ze zmiennych (private) w klasie Particle i ustalanych w momencie tworzenia cząstki. Przekazywanie koloru do klasy Applet_walk poprzez metodę get_color.

Przenieś rysowanie do klasy dziedziczącej po klasie Canvas. (W klasie Canvas zdefiniowana jest metoda paint, którą przeddefiniujemy.) Następnie dodaj taki obiekt do apletu. Dodaj linie brzegowe płótna.

Rozwiązanie:

```

import java.awt.*;
class Wykres extends Canvas{
    int rozmiar;
    public Wykres(int wielkosc){
        setSize(wielkosc,wielkosc);
        rozmiar=wielkosc;
        setBackground(Color.orange);
    }
    public void paint(Graphics g){
        int i,n_step;
        n_step=1000;
        g.drawLine(0, 0, rozmiar, 0);
        Walkerd2 w1 = new Walkerd2(rozmiar/3,rozmiar/3);
        for (i=1;i<=n_step;i++){
            w1.step();
            g.setColor(new Color (255,0,0));

```

```

        g.drawRect(w1.get_x(),w1.get_y(),1,1);
    }
}

```

```

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Label;

public class Applet_walk_canvas extends Applet{
    int applet_width=500;
    int applet_height=500;
    public void init(){
        setSize(applet_width,applet_height);
        Wykres wyk = new Wykres(200);
        add(new Label("błądzenie przypadkowe"));
        add(wykr);
    }
} // koniec Applet_walk

```

Dodaj periodyczne warunki brzegowe: cząstka wychodząc poza obszar, wchodzi do niego z drugiej strony.

Zmień rozkład apletu na BorderLayout. Płótno dodaj do części środkowej a napis „błądzenie przypadkowe” do północnej.

Obsługa zdarzeń i komponenty AWT

Słuchacz to obiekt, który może obsługiwać zdarzenia (wciśnięcie guzika, przesunięcie myszki lub suwaka, wpisanie tekstu, itp.). Aby móc generować obiekty-Słuchaczy, klasa musi implementować interfejs nasłuchu. Tzn., metody interfejsu nasłuchu muszą zyskać konkretne definicje.

Zdarzenia przytrafiają się obiektom. Obiekt, któremu przytrafiło się zdarzenie nazywany jest źródłem (*source*). Zdarzenie do obsługi przekazywane jest ze źródła do słuchacza tylko wtedy, gdy do danego źródła przyłączony jest słuchacz reagujący na ten typ zdarzenia. Przyłączenia dokonujemy za pomocą odwołania:

```
z.addxxxListener(l);
```

gdzie *z* - źródło, *xxx* - rodzaj nasłuchiowanych zdarzeń (*Action*, *Item*, *Adjustement*, *MouseMotion*), *l* - słuchacz.

ActionEvent to klasa zdarzeń oznaczających wykonanie akcji. Słuchacz akcji (obiekt klasy, która ma ją obsługiwać) musi implementować interfejs

ActionListener i zdefiniować metodę *actionPerformed* tego interfejsu.

AWT - Abstract Windowing Toolkit

Jest to zbiór klas zapewniających tworzenie prostych elementów graficznego interfejsu użytkownika (*Applet*, *Label*, *Button*, *TextField*, ...). Znajdują się one w pakiecie *java.awt* (wymagany *import*).

Konwersja temperatury - Aplet

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class TempApplet1 extends Applet implements ActionListener {
    TextField tFahr;
    Label lCent;
    public void init(){
        tFahr=new TextField(10);
        lCent=new Label ("I'll tell you what that is in degrees C");
        add (new Label("Please type the temperature (deg F)"));
        add(tFahr);
        add(lCent);
        tFahr.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e){
        int fahr=0,
            cent=0;
        fahr=Integer.parseInt(tFahr.getText());
        cent=(int)(5.0*(fahr-32)/9.0);
        lCent.setText(fahr+" deg F is "+cent+" deg C");
    }
}
```