

Język Java

Zestaw 3

2013 / 2014

Klasy proszę umieścić w pakiecie `mandelbrot`.

1. **Complex**

Proszę zaimplementować klasę obsługującą liczby zespolone:

```
public class Complex
{
    private double r, i;

    Complex();
    Complex(double);
    Complex(double, double);
    Complex(Complex);
    Complex(String);

    static Complex Add(Complex, Complex);
    /* Zwraca sumę dwóch liczb zespolonych */
    static Complex Sub(Complex, Complex);
    /* Zwraca różnicę dwóch liczb zespolonych */
    static Complex Mul(Complex, Complex);
    /* Zwraca iloczyn dwóch liczb zespolonych */
    static Complex Div(Complex, Complex);
    /* Zwraca iloraz dwóch liczb zespolonych */
    static double Abs(Complex);
    /* Zwraca moduł liczby zespolonej */
    static double Phase(Complex);
    /* Zwraca fazę liczby zespolonej */
    static double SqrAbs(Complex);
    /* Zwraca kwadrat modułu liczby zespolonej */
    static double Re(Complex);
    /* Zwraca część rzeczywistą liczby zespolonej */
    static double Im(Complex);
    /* Zwraca część urojoną liczby zespolonej */

    /* Poniższe metody modyfikują aktualny obiekt i zwracają 'this' */
    Complex Add(Complex); // Dodaje liczbę zespoloną
    Complex Sub(Complex); // Odejmuje liczbę zespoloną
    Complex Mul(Complex); // Mnoży przez liczbę zespoloną
    Complex Div(Complex); // Dzieli przez liczbę zespoloną

    double Abs();
    double SqrAbs();
    double Re();
}
```

```

    double Im();

    String toString();
    /* Zwraca String z zapisaną
       liczbą zespoloną formacie "-1.23+4.56i" */

    static Complex valueOf(String);
    /* Zwraca liczbę zespoloną o wartości podanej
       w argumencie w formacie "=1.23+4.56i" */

    void setRe(double);
    /* Przypisuje podaną wartość części rzeczywistej */

    void setIm(double);
    /* Przypisuje podaną wartość części urojonej */

    void setVal(Complex);
    /* Przypisuje podaną wartość */
}

```

Powyższe metody i konstruktory proszę zadeklarować jako publiczne.
 Przydatne klasy: *Math*

2. Mandelbrot

Proszę zapoznać się z definicją zbioru Mandelbrota:

http://pl.wikipedia.org/wiki/Zbiór_Mandelbrota

i napisać program, który ten zbiór rysuje. Wykorzystać własną klasę *Complex*. Kolorem czarnym proszę rysować te punkty p w płaszczyźnie zespolonej, dla której ciąg $z_{n+1} = z_n^2 + p$ nie dąży do ∞ . Dla pozostałych punktów, kolor uzależnić od prędkości ucieczki (jak szybko, czyli minimalne n , dla którego $SqrAbs(z_n)$ jest większe od pewnego progu r^2 - ma to być parametr, domyślnie np. 4). Proszę korzystać z formatu RGB i wymyśleć "ładną" funkcję kolorującą. Ograniczyć się do maksymalnie ok. 100 iteracji ciągu z_n , aby określić czy ciąg jest rozbieżny.

Proszę zapoznać się z klasą *BufferedImage* z pakietu *java.awt.image*, która przechowuje dane obrazu.

Sugeruję wybrać `imageType = TYPE_INT_RGB`.

Aby postawić punkt o danym kolorze można użyć metody `setRGB`.

Proszę napisać klasę *ImagePanel*, która dziedziczy po klasie *JPanel* oraz implementuje interfejs *MouseMotionListener*. Proszę przeciążyć metodę *paint(Graphics)*, tak aby rysowała tło, które jest obrazem (np. typu *BufferedImage*). Klasa ta ma umożliwiać wybór prostokątnego obszaru (obrazka) przy pomocy myszki, należy odpowiednio zaimplementować metodę *mouseDragged*. Proszę zwrócić uwagę na metodę *setXORMode* klasy *Graphics*. Po wybraniu obszaru proszę wywołać metodę *regionChosen* obiektu klasy, która implementuje interfejs *RegionChooser*. Obiekt ten może zostać przekazany np. jako argument konstruktora *ImagePanel*. *ImagePanel* ma rysować szare tło gdy nie wybrano żadnego obrazu (null).

Proszę napisać interfejs **RegionChooser**.

Napisać klasę **Mandelbrot**, która dziedziczy po klasie **JFrame** oraz implementuje interfejs *WindowListener*, *ActionListener* oraz własny interfejs **RegionChooser**.

Proszę użyć "układacza" *GridBagLayout* wywołując metodę *setLayout* oraz wykorzystać klasę *GridBagConstraints*.

Nowe komponenty (przyciski, pola tekstowe, etc.) dodajemy wywołując metodę *add*. Proszę zwrócić uwagę na metody *actionPerformed* interfejsu *ActionListener*, *windowClosing* interfejsu *WindowListener*, *addWindowListener*, *dispose* dla *Frame*, *addActionListener* dla *Button* oraz *getSource* dla *ActionEvent*

Dodać pole umożliwiające wybór: zakresu obrazka (czyli zakres p), parametru r , wielkość obrazka (liczba pixeli (*width* i *height*)), przycisk rysujący fraktal. Dodać pole w której znajdują rysunek fraktala czyli **ImagePanel**. Umożliwić wybór zakresu obrazka myszką.

Podsumowanie:

Klasa **ImagePanel** ma rysować obraz i umożliwiać wybór prostokątnego obszaru przy pomocy myszki, wtedy wywołuje metodę **regionChoosen** klasy **Mandelbrot**.

Klasa **Mandelbrot** ma za zadanie zebrać parametry fraktala, wygenerować (policzyć) obrazek typu *BufferedImage* z fraktalem, uaktualnić **ImagePanel** aby wyświetlił policzony obraz.

Przydatne klasy: *Frame*, *Panel*, *TextField*, *Label*, *Button*, *GridBagLayout*, *BufferedImage* oraz **Complex**.

Andrzej Görlich
andrzej.goerlich@uj.edu.pl
<http://th.if.uj.edu.pl/~atg/Java>