Programowanie w języku Java

dr Przemysław Juszczuk

Wykład 3

```
6   package javaapplication70;
7
8   /**
9    *
10   * @author Scorpu
11   */
12  public class Test
13  {
14      int val1;
15      int val2;
16
17      public Test()
18      {
19          this.val1 = 0;
20          this.val2 = 1;
21      }
22
23      public Test(int i, int j)
24      {
25          this.val1 = i;
26          this.val2 = j;
27      }
28
29      public void Metoda()
30      {
31          System.out.println("Tutaj przykładowa metoda");
32      }
```

Rysunek: Dziedziczenie 1

```
12  public class NewTest extends Test
13  {
        int val1;
        int val2;
16      int val3;
17
18      public NewTest(int vol1, int vol2, int vol3)
19      {
20          super();
21          val3 = 5;
22      }
23
24      @Override
        public void Metoda()
26      {
27          System.out.println("Nadpisana metoda");
28      }
29  }
```

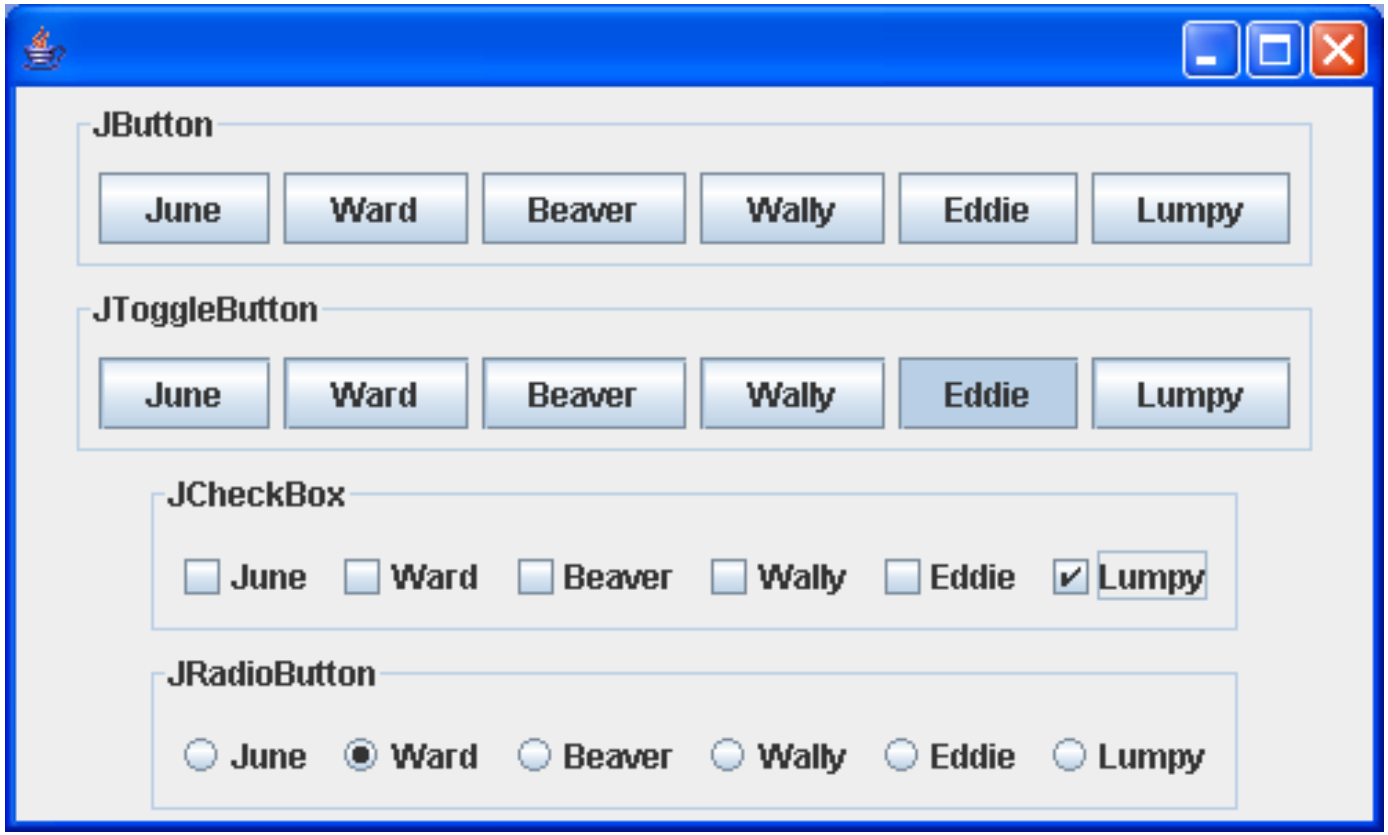
Rysunek: Dziedziczenie 2

```
12  public class JavaApplication70 {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18
19          Test obiekt_1 = new Test();
20          NewTest obiekt_2 = new NewTest(2,3,5);
21
22          obiekt_1.Metoda();
23          obiekt_2.Metoda();
24      }
25
26  }
```

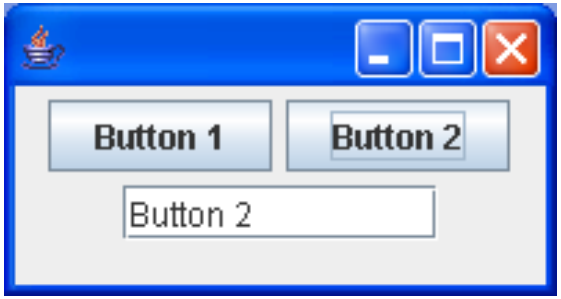
Rysunek: Adnotacja Override i nadpisanie metody z klasy bazowej

Elementy Swing

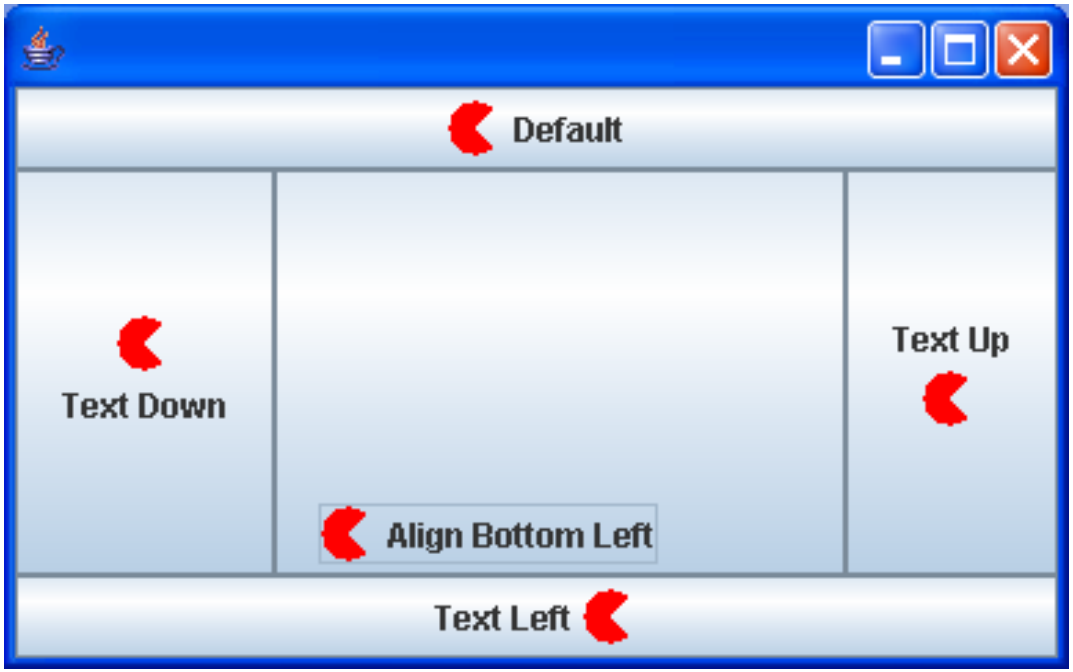
- wszystkie elementy graficzne (przyciski, kontrolki) rysowane są od zera z poziomu Javy;
- ten sam wygląd i zachowanie na wszystkich platformach ze wsparciem Javy;
- brak efektów graficznych związanych z konkretnym systemem operacyjnym;
- możliwość symulowania wyglądu właściwego np. dla komponentów systemu Windows.



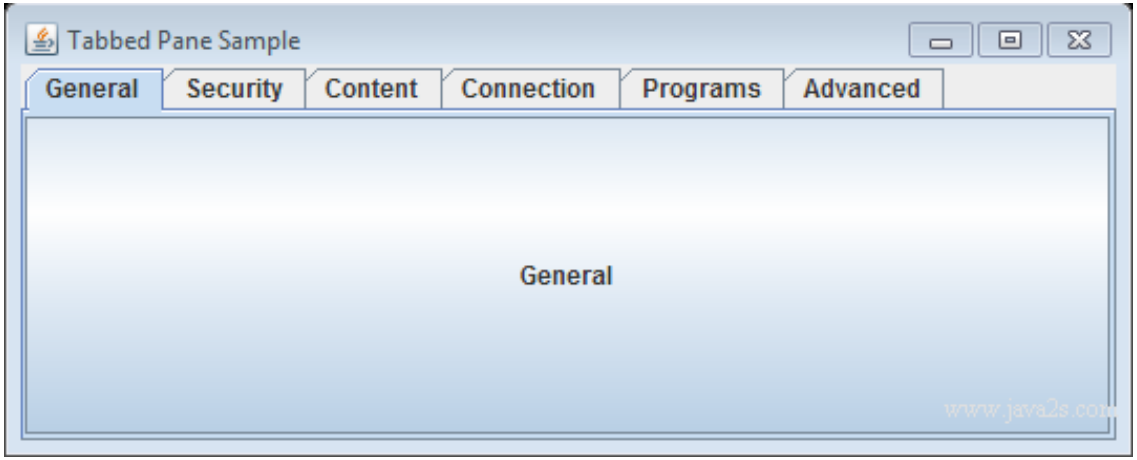
Rysunek: Przyciski i ich rodzaje



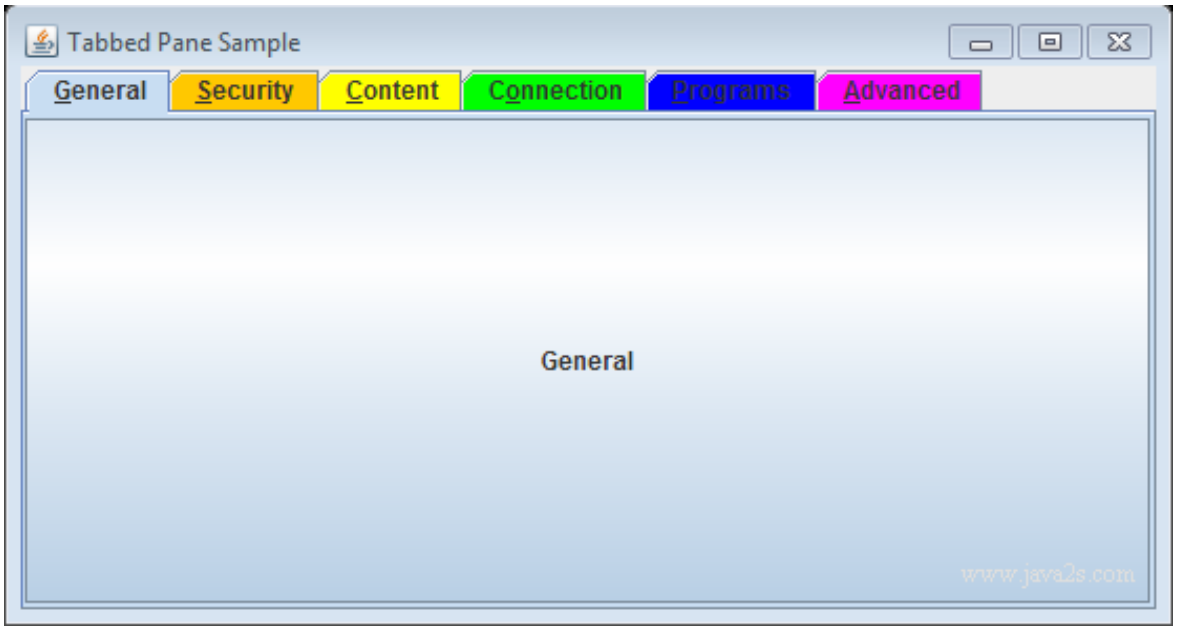
Rysunek: Focus na przycisku i reakcja na naciśnięcie



Rysunek: Komponenty i ich rozmieszczenie



Rysunek: Komponent TabPane



Rysunek: Komponent TabPane i kolory

Komponenty związane z wyborem

- ComboBox;
- List;
- Button Group;
- Spinner;
- Tree (opcjonalnie).

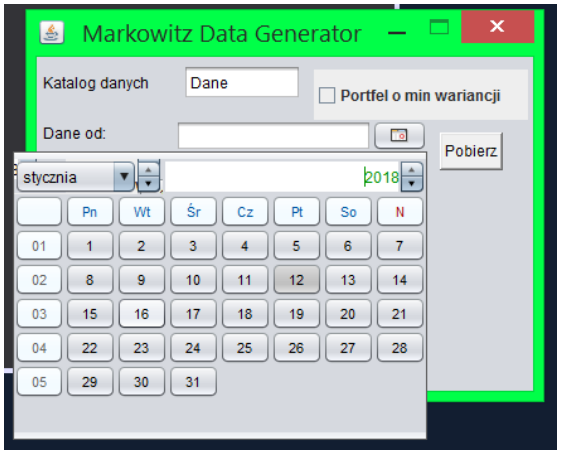
Wykorzystanie powyższych elementów wymaga zbudowania pewnego słownika na początek. Może to być stały zestaw, którym komponenty są inicjalizowane podczas uruchomienia aplikacji. Alternatywnie wartości mogą być dobierane dynamicznie np. jako:

- wartości z pliku;
- elementy zdefiniowane wcześniej przez użytkownika;
- wartości pobrane z bazy danych;
- inne, zewnętrzne źródło, jak np. strona internetowa.

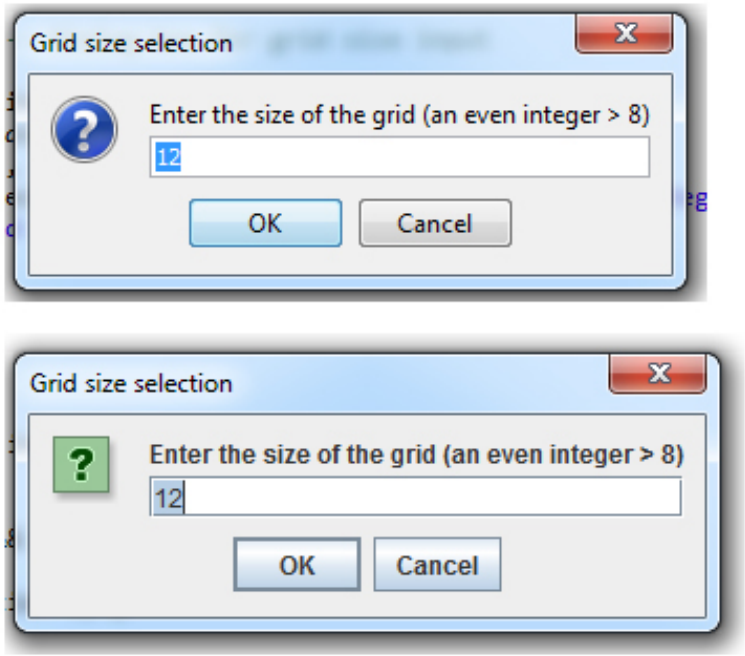
Okna Swing

Wykorzystanie gotowych okien, jako elementów składowych całej aplikacji:

- File Chooser;
- Color Chooser;
- Option Pane*;
- Calendar.



Rysunek: Przykład użycia kalendarza



Rysunek: Interakcja z użytkownikiem

Interakcja

- użytkownik jest najsłabszym ogniwem (w działającym systemie);
- zawsze musimy zakładać, że użytkownik kliknie/ wybierze coś dziwnego;
- "Podaj liczbę – aaa";
- "Podaj dwie daty, pierwsza powinna poprzedzać drugą: 10 stycznia 2020, 5.12.1999"

Interakcja II

- ograniczamy możliwość wprowadzania danych do minimum;
- dajemy jasno określony wybór;
- pamiętamy o różnych formatach dat, różnych strefach czasowych, różnych walutach, różnych formatowaniach liczb (przecinek zamiast kropki w liczbie);
- zapisujemy dane już wprowadzone przez użytkownika w sytuacji, kiedy do wypełnienia jest długi formularz;
- dodajemy dymki z podpowiedzią.

Komponenty menu

- ograniczamy do minimum liczbę wyskakujących okien;
- ograniczamy liczbę wyskakujących alertów;
- elementy wewnątrz menu posortowane alfabetycznie;
- przyciski (w szczególności typu RadioButton) pogrupowane;
- zapamiętujemy wybory użytkownika (np. zaznaczone checkboxy);
- pasek menu w domyślnym miejscu (lewy górny róg).

Elementy GUI a efekty wizualne

- gradient tła – pozwala podkreślić elementy znajdujące się na pierwszym planie;
- dodanie tła w aplikacji nie powinno mieć żadnych wyraźnych elementów przyciągających wzrok (chyba, że chodzi np. o logo, które jest jednocześnie łączem);
- cały interfejs bazuje na mocno ograniczonej liczbie kolorów o różnej jasności i nasyceniu;
- stylizowane rogi są dobre do podkreślenia podgrup – np grupa przycisków zawierająca podgrupę RadioButton;
- pozwalaj użytkownikowi na dobór palety kolorów w ustawieniach aplikacji.

Elementy GUI a zdarzenia

- elementy GUI generują zdarzenia;
- rodzaj zdarzenia opisywany jest przez NazwaEvent (ActionEvent, MouseEvent, KeyEvent);
- kod obsługi zdarzenia powiązany jest z interfejsem NazwaListener (ActionListener, KeyListener);
- w aplikacjach mobilnych jest nieco łatwiej – wystarczy obsługa zdarzenia OnClick;
- z poziomu kodu w Java łatwo zrozumieć cały mechanizm:
  1. przycisk.addActionListener(sluchacz);
  2. ActionListener sluchacz = new ActionListener()
  3. public void actionPerformed(ActionEvent z)
- komponent powiązany jest ze słuchaczem, który jest obiektem implementującym okresloną metodę.

Rodzaje interfejsów obserwatorów

- ActionListener – kliknięcie przycisku;
- MouseListener – operacje wykonywane myszką (mouseClicked, mouseReleased, mouseEntered, mouseReleased) – np. zmiana tekstu po najechaniu myszki (lub zmiana w hiperłącze);
- MouseMotionListener – ruch kursora myszy nad komponentem;
- KeyListener – wciśnęcie przycisku na klawiaturze;
- TextListener – zmiana tekstu wewnątrz określonego elementu;
- WindowListener – zdarzenia związane z oknem – minimalizacja, maksymalizacja, okno aktywne / okno nieaktywne.

*Wątki a GUI

- kiedy nie mamy API do strony;
- kiedy nie mamy ustalonego formatu pobierania danych (często pojawia się JSON – w obsłudze zbliżony do XML);
- konieczność pobierania danych bezpośrednio ze strony;
- zbyt częste odpytywanie serwera może grozić banem;
- zbyt częste odpytywanie serwera w jednym wątku blokuje całą aplikację;
- wątki do wykonywania równolegle tej samej akcji – najczęściej pojawia się przykład z paskiem postępu działającym niezależnie od całej aplikacji;
- a co z połączeniem z serwerem w tle?

```java
@Override
public void run()
{
    try
    {
        PobierajDane();
        WypiszDane();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
```

Rysunek: Implementacja metody Run – wymuszenie implementacji poprzez implementację interfejsu Runnable

```
public static void main(String[] args) {

    int odczyt = 3;

    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {

            //start the simulator thread now
            new BittrexEA();
            Executors.newScheduledThreadPool(1).scheduleAtFixedRate(new Simulator(), 2, odczyt, TimeUnit.SECONDS);

        }
    });

}
```

Rysunek: Odpalenie wątków do pobierania danych

Zagadnienia

- Przykład z wątkami i JSON
- Excel i obsługa plików xls
- Generowanie i odczyt plików pdf
- Generowanie skryptów do aplikacji zewnętrznych
- Zagadnienia do egzaminu

```
6   package bittrexea;
7
8   import java.io.BufferedReader;
9   import java.io.InputStream;
10  import java.io.InputStreamReader;
11  import java.io.Reader;
12  import java.net.URL;
13  import java.nio.charset.Charset;
14  import java.util.ArrayList;
15  import org.json.JSONArray;
16  import org.json.JSONObject;
17
```

Rysunek: Import elementów związanych z formatem JSON

```
22  public class Simulator implements Runnable  // klasa na wątkach do pobierania
23  {
24
25      public static String[] rynki;  // tutaj trzymamy wszystkie nazwy kryptowalut
26
27      public static ArrayList<Double>[] dane;  // tutaj trzymamy dane tickowe
28                                               // pierwszemu elementowi z tablicy
29
30      public Simulator()  // w konstruktorze sprawdzamy, ile jest aktywnych rynków
31      {
            boolean active = false;
            String market = "";
34          int indeks = 0;
35          try
36          {
37             JSONObject json = readJsonFromUrl("https://bittrex.com/api/v1.1/public/getmarkets");  // pobieramy objekty
38             JSONObject selectedMarket;
39           JSONArray jsonArray = json.getJSONArray("result"); // przerzucamy JSON do tablicy
40           rynki = new String[jsonArray.length()];    // wiemy już, ile jest krypto na rynku w danym momencie
41           dane = new ArrayList[jsonArray.length()];
```

Rysunek: Konstruktor i implementacja interfejsu Runnable

Response Example (200 OK)


```
{
  "success": true,
  "message": "''",
  "result": [
    {
      "MarketCurrency": "LTC",
      "BaseCurrency": "BTC",
      "MarketCurrencyLong": "Litecoin",
      "BaseCurrencyLong": "Bitcoin",
      "MinTradeSize": 0.01,
      "MarketName": "BTC-LTC",
      "IsActive": true,
      "IsRestricted": false,
      "Created": "2014-02-13T00:00:00",
      "Notice": "BTC-LTC",
      "IsSponsored": false,
      "LogoUrl":
"https://storage.blob.core.windows.net/public/8637ccad-9e7f-45ac-
8f03-a41b440e3911.png"
    }
  ]
}
```


Rysunek: Przykład formatu JSON – bittrex

Response Example (200 OK)


```
{
  "success": true,
  "message": "''",
  "result": [
    {
      "Bid": 2.05670368,
      "Ask": 3.35579531,
      "Last": 3.35579531
    }
  ]
}
```


Rysunek: Przykład formatu JSON – bittrex

GET /public/getticker

DESCRIPTION

Used to get the current tick values for a market.

REQUEST PARAMETERS

market: string	REQUIRED	a string literal for the market (ex: BTC-LTC)
	in query	

Rysunek: Przykład formatu JSON – bittrex

Response Example (200 OK)


```
{
  "success": true,
  "message": "''",
  "result": [
    {
      "MarketName": "BTC-LTC",
      "High": 0.0135,
      "Low": 0.012,
      "Volume": 3833.97619253,
      "Last": 0.01349998,
      "BaseVolume": 47.03987026,
      "TimeStamp": "2014-07-09T07:22:16.72",
      "Bid": 0.01271001,
      "Ask": 0.012911,
      "OpenBuyOrders": 45,
      "OpenSellOrders": 45,
      "PrevDay": 0.01229501,
      "Created": "2014-02-13T00:00:00",
      "DisplayMarketName": "string"
    }
  ]
}
```


Rysunek: Przykład formatu JSON – bittrex

```
for (int i = 0 ; i < dane.length ; i++)
{
    dane[i] = new ArrayList<>();    // dane są w talbicach dynamicznych, więc tworzymy tablice
}

for (int i =0 ; i < jsonArray.length() ; i++)
   {
      selectedMarket = (JSONObject)jsonArray.get(i);
       active = (boolean)selectedMarket.get("IsActive");  // sprawdzamy, czy dana krypto jest aktywna
       market = (String)selectedMarket.get("MarketName");  // i zapisujemy jej nazwe do tablicy
       System.out.println(market);
      if (active == true)
       {
        rynki[indeks] = market;
        indeks++;
       }
   }
```

Rysunek: Format JSON

```java
@Override
public void run()
{
    try
    {
        PobierajDane();
        WypiszDane();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
```

Rysunek: Implementacja metody Run()

```
public static void PobierajDane() throws Exception  // druga metoda odpalona w tle, tym razem do pobierania danych
{
    try
    {
     JSONObject json = readJsonFromUrl("https://bittrex.com/api/v1.1/public/getmarketsummaries");  // tutaj mamy info o rynkach, wolumen oraz cenę
     JSONObject selectedMarket;
     JSONArray jsonArray = json.getJSONArray("result");
     rynki = new String[jsonArray.length()];
     Double dat;
     for (int i =0 ; i < jsonArray.length() ; i++)   // pobieramy wszystkie dostępne krypto
        {
            selectedMarket = (JSONObject)jsonArray.get(i);  // wyciągamy info o jednej krypto
            dat = (Double)selectedMarket.getDouble("Last");  // i pobieramy ostatnią cenę (możemy zastąpić przez Bid lub Ask oraz pobierać też wolumen)
            dane[i].add(dat);  // wrzucamy do tablicy dynamicznej - czyli przy każdym odczycie zwiększa nam się liczba elementów tablicy o 1
            rynki[i] = selectedMarket.getString("MarketName");;
        }

    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
```

Rysunek: Metoda pobierania danych na wątkach

{"success":true,"message":"","result":[{"MarketName":"BTC-RVN","High":0.00000310,"Low":0.00000301,"Volume":2493956.88670675,"Last":0.00000303,"BaseVolume":7.61734971,"Tim
23T19:22:00.29","Bid":0.00000302,"Ask":0.00000303,"OpenBuyOrders":418,"OpenSellOrders":4444,"PrevDay":0.00000308,"Created":"2018-08-13T22:11:25.473"},{"MarketName":"BTC-
VITE","High":0.00000138,"Low":0.00000138,"Volume":377.35849057,"Last":0.00000138,"BaseVolume":0.00052075,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00000135,"Ask":0.00000144,"OpenBuyOrders":30,"OpenSellOrders":86,"PrevDay":0.00000138,"Created":"2019-01-15T17:19:32.057"},{"MarketName":"BTC-
DRGN","High":0.00000465,"Low":0.00000435,"Volume":20686.64955669,"Last":0.00000435,"BaseVolume":0.09280758,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00000426,"Ask":0.00000436,"OpenBuyOrders":69,"OpenSellOrders":244,"PrevDay":0.00000465,"Created":"2019-01-08T17:05:55.26"},{"MarketName":"BTC-
SNT","High":0.00000124,"Low":0.00000121,"Volume":297462.09817427,"Last":0.00000123,"BaseVolume":0.36451085,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00000121,"Ask":0.00000123,"OpenBuyOrders":43,"OpenSellOrders":1049,"PrevDay":0.00000124,"Created":"2017-06-28T18:26:15.73"},{"MarketName":"USD-
ZEN","High":10.32515000,"Low":9.50000000,"Volume":1093.33620141,"Last":9.60015000,"BaseVolume":10809.91062300,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":9.70000000,"Ask":9.81800000,"OpenBuyOrders":176,"OpenSellOrders":140,"PrevDay":10.29261000,"Created":"2019-01-31T18:29:51.89"},{"MarketName":"BTC-
VBK","High":0.00000074,"Low":0.00000057,"Volume":1685194.13513333,"Last":0.00000061,"BaseVolume":1.09667444,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00000060,"Ask":0.00000062,"OpenBuyOrders":98,"OpenSellOrders":1167,"PrevDay":0.00000065,"Created":"2019-04-05T01:01:47.63"},{"MarketName":"BTC-
FLDC","High":0.00000003,"Low":0.00000003,"Volume":2790235.36877944,"Last":0.00000003,"BaseVolume":0.08370704,"TimeStamp":"2019-10-
29T18:10:31.6","Bid":0.00000003,"Ask":0.00000004,"OpenBuyOrders":45,"OpenSellOrders":1588,"PrevDay":0.00000003,"Created":"2015-02-23T23:01:53.92"},{"MarketName":"USD-
XRP","High":0.23664000,"Low":0.22100000,"Volume":455311.01513829,"Last":0.22145000,"BaseVolume":103663.27712842,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.22257000,"Ask":0.22316000,"OpenBuyOrders":396,"OpenSellOrders":2348,"PrevDay":0.23485000,"Created":"2018-08-20T21:47:01.797"},{"MarketName":"USD-
ETC","High":9.49588000,"Low":8.19300000,"Volume":3939.68906057,"Last":8.29639000,"BaseVolume":34404.34775528,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":8.23738000,"Ask":8.30826000,"OpenBuyOrders":242,"OpenSellOrders":371,"PrevDay":9.10247000,"Created":"2018-08-20T21:47:05.61"},{"MarketName":"ETH-
DMT","High":0.00082662,"Low":0.00082662,"Volume":13.60075169,"Last":0.00082662,"BaseVolume":0.01124265,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00075408,"Ask":0.00094000,"OpenBuyOrders":33,"OpenSellOrders":58,"PrevDay":0.00082662,"Created":"2018-03-27T17:20:49.18"},{"MarketName":"BTC-
XST","High":0.00000732,"Low":0.00000677,"Volume":57134.69604023,"Last":0.00000720,"BaseVolume":0.40472909,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00000711,"Ask":0.00000734,"OpenBuyOrders":143,"OpenSellOrders":734,"PrevDay":0.00000683,"Created":"2014-07-09T20:33:00.59"},{"MarketName":"BTC-
XDN","High":0.00000008,"Low":0.00000007,"Volume":1017451.56418969,"Last":0.00000007,"BaseVolume":0.07175933,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00000007,"Ask":0.00000008,"OpenBuyOrders":144,"OpenSellOrders":2244,"PrevDay":0.00000007,"Created":"2014-06-21T18:37:57.64"},{"MarketName":"BTC-
MONA","High":0.00014189,"Low":0.00013498,"Volume":15053.04604571,"Last":0.00013624,"BaseVolume":2.09236639,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00013635,"Ask":0.00013830,"OpenBuyOrders":203,"OpenSellOrders":1273,"PrevDay":0.00014140,"Created":"2014-04-18T22:00:00"},{"MarketName":"BTC-
AERGO","High":0.00000000,"Low":0.00000000,"Volume":0.00000000,"Last":0.00000000,"BaseVolume":0.00000000,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00000105,"Ask":0.00000669,"OpenBuyOrders":2,"OpenSellOrders":115,"PrevDay":0.00000000,"Created":"2019-03-14T22:46:31.223"},{"MarketName":"ETH-
TUSD","High":0.00628997,"Low":0.00601368,"Volume":17201.94877326,"Last":0.00628997,"BaseVolume":106.09036450,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00620137,"Ask":0.00628985,"OpenBuyOrders":141,"OpenSellOrders":67,"PrevDay":0.00602010,"Created":"2018-03-19T17:38:47.76"},{"MarketName":"BTC-
OK","High":0.00000355,"Low":0.00000267,"Volume":312952.41760836,"Last":0.00000310,"BaseVolume":0.95129010,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":0.00000316,"Ask":0.00000331,"OpenBuyOrders":132,"OpenSellOrders":1432,"PrevDay":0.00000295,"Created":"2015-05-30T19:26:37.523"},{"MarketName":"USD-
ZEC","High":51.94500000,"Low":48.03400000,"Volume":513.84918961,"Last":48.46900000,"BaseVolume":25648.81823190,"TimeStamp":"2020-01-
23T19:22:00.29","Bid":48.18100000,"Ask":48.76800000,"OpenBuyOrders":124,"OpenSellOrders":314,"PrevDay":51.40500000,"Created":"2018-09-05T21:34:02.11"},{"MarketName":"BTC-

Rysunek: Struktura danych w pliku

```
public static JSONObject readJsonFromUrl(String url) throws Exception
{
  InputStream is = new URL(url).openStream();
  try {
    BufferedReader rd = new BufferedReader(new InputStreamReader(is, Charset.forName("UTF-8")));
    String jsonText = readAll(rd);
    JSONObject json = new JSONObject(jsonText);
    return json;
  } finally {
    is.close();
  }
}
```

Rysunek: Odczyt danych

```
public static void main(String[] args) {

    int odczyt = 3;

    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {

            //start the simulator thread now
            new BittrexEA();
            Executors.newScheduledThreadPool(1).
                    scheduleAtFixedRate(new Simulator(), 2, odczyt, TimeUnit.SECONDS);

        }
    });

}
```

Rysunek: Działanie na wątkach

Jexcel API – wstęp

- odczyt i obsługa plików w formacie Excel;
- obsługa wielu arkuszy w jednym pliku;
- formatowanie komórek;
- odczyt i modyfikacja istniejących arkuszy;
- przetwarzanie formuł w wielu językach;
- obsługa kopiowania i wstawiania wykresów.

```
PrintWriter writer = new PrintWriter("results.txt", "UTF-8");


double[][][] data;    // pełene dane: 1 wymiar to miesiąc, drugi to kryterium, trzeci to fundusz
double[][][] fund_data;  // wszystkie dane ale tylko dla aktywnych kryteriów
double[][] RoR_data;   // dane tylko dla pierwszego kryterium (czyli domyślnie Rate of Return

data = ReadData(nazwa,ile_arkuszy, ile_kryteriow, ile_funduszy);    // wczytanie wszystkich danych
fund_data = CopyActiveCriteria(data, aktywne, ile_arkuszy, ile_kryteriow, ile_funduszy);
RoR_data = CopyRoRData(fund_data, aktywne, ile_arkuszy, ile_funduszy);
```

Rysunek: Struktura danych

1		1 month	3 months	6 months	1 year	2 years	3 years	5 years	since now	10 years	3 years	3 years	3 years	3 years
2	Templeton Growth Fund;A	3,77	8,34	4,36	-0,51	21,82	49,68	61,58	5,29	66,63	11,82	1,14	0,3	0,94
3	Templeton World Fund;A	4	9,05	3,92	-0,1	21,24	44,72	59,26	5,76	94,03	11,44	1,12	0,28	0,88
4	Templeton Global Opportunities Trust;A	4,71	9,45	4,41	1	19,76	43,25	48,42	6,9	97,67	11,09	0,84	0,27	0,88
5	Aberdeen Global - World Equity A2 Acc	3,68	4,48	0,4	-1,59	7,67	21,97	42,5	3,1	103,66	9,37	0,94	0,17	0,56
6	AllianceBernstein-Global Gth Trends Pf	2,21	7,73	5,87	9,62	24,1	31,81	40,03	7,5	29,6	10,44	1,07	0,22	0,7
7	Pinebridge Global Funds - Global Focu	1,26	4,86	5,77	10,15	20,46	38,46	44,36	4,07	66,67	10,22	0,94	0,29	0,94
8	Threadneedle (Lux)-Global Focus AU	0,38	4,88	4,73	9,8	27,4	41,67	63,37	2,6	107,72	11	1,04	0,28	0,93
9	Investec GSF Global Dynamic A Acc	2,54	9,47	6,84	8,24	24,95	42,65	58,51	7,95	99,99	11,06	1,14	0,27	0,85
10	Investec GSF Global Strategic Equity A	3,57	9,17	8,25	10,53	33,01	53,41	68,26	8,4	98,18	10,9	1,11	0,34	1,06
11	JPMorgan Global Unconstrained Equity	4,17	9,45	7,78	9,78	28,64	43,55	47,82	6,13	61,19	11,07	1,05	0,29	0,94
12	Manulife Global Fund-International Gro	2,28	5,97	6,78	6,92	26,17	44,96	54,75	4,12	66,57	10,17	0,99	0,33	1,02
13	MFS Meridian Funds Global Equity A1	-3,08	6,23	17,52	30,54	43,79	71,65	98,08	11,5	144,83	10,94	1,12	0,59	1,33
14	Templeton Global A (Ydis) USD	5,07	10,49	6,59	1,72	27,51	57,17	57,37	8,06	68,06	12,09	1,2	0,32	1,04
15	Templeton Global (Euro) A (Ydis) EUR	0,88	11,57	20,33	27,38	51,19	83,85	82,57	17,1	81	12,47	1,04	0,65	1,62
16	United International Growth Fund	-1,89	2,31	3,91	9,87	24,97	34,96	28,44	2,01	27,37	9,82	1	0,31	0,78
17	AXA Rosenberg Global Equity Alpha A	1,7	5,35	4,46	5,63	26,62	46,39	64,77	2,82	76,12	10,26	1,03	0,33	1,01
18	Jupiter Ecology	-2,58	3,63	8,38	8,75	20,62	40,3	39,24	4,89	134,44	10,19	0,79	0,36	1,02
19	Jupiter Global Managed Inc	0,06	3,51	10,7	17,35	21,57	48,28	51,34	5,55	176,13	9,51	0,89	0,44	1,16
20	AllianceBernstein-Global Gth Trends Pf	2,28	7,94	6,29	10,51	26,09	35,01	45,75	7,78	40,41	10,44	1,07	0,24	0,76
21	AllianceBernstein-Global Gth Trends Pf	2,13	7,46	5,34	8,54	21,64	27,93	33,24	7,14	17,29	10,43	1,07	0,19	0,62
22	AllianceBernstein-Global Gth Trends Pf	2,17	7,6	5,64	9,13	23	30,05	36,94	7,33	23,91	10,43	1,07	0,21	0,66

Rysunek: Struktura danych – plik

STOPY ZWROTU i PROWIZJE

Całość | 1M | 3M | 1R | 3L | 5L | Sygnały

okres	stopa zwrotu
dzień	-0.38%
miesiąc	2.02%
kwartał	4.72%
rok	3.31%
trzy lata	5.22%
pięć lat	12.83%

Rysunek: Dane funduszy

FUNDUSZE POKREWNE

Inne Fundusze Tej Firmy

Nazwa	Data	Wycena	Ost. Zm.	1M	3M	6M	12M	24M	36M	
AEGON UFK Akcji Rosyjskich	2020-01-20	1.6 PLN	0.87%	8.33%	19.60%	21.34%	38.92%	49.44%	41.72%	
AEGON UFK Akcji Tureckich	2020-01-20	0.8 PLN	1.19%	11.76%	20.53%	14.26%	22.23%	-17.49%	-4.09%	
AEGON UFK Alfa ETF 1 (PLN)	2020-01-21	13.3 PLN	-0.52%	1.37%	3.33%	1.75%	5.37%	-3.19%	11.63%	
AEGON UFK Alfa ETF 10 (PLN)	2020-01-21	11.3 PLN	0.09%	-0.53%	-0.44%	0.27%	2.73%	3.11%	9.73%	
AEGON UFK Alfa ETF 2 (PLN)	2020-01-21	12.8 PLN	-0.39%	0.87%	2.56%	1.19%	4.74%	-2.88%	11.78%	
AEGON UFK Alfa ETF 3 (PLN)	2020-01-21	11.7 PLN	-0.34%	0.69%	2.18%	1.12%	4.46%	-2.66%	10.69%	
AEGON UFK Alfa ETF 4 (PLN)	2020-01-21	11.4 PLN	-0.26%	0.62%	1.79%	1.06%	4.40%	-2.06%	11.67%	
AEGON UFK Alfa ETF 5 (PLN)	2020-01-21	11.8 PLN	-0.26%	0.43%	1.38%	1.03%	4.26%	-0.68%	10.43%	
AEGON UFK Alfa ETF 6 (PLN)	2020-01-21	11.1 PLN	-0.18%	0.09%	1.00%	0.91%	4.13%	0.54%	11.91%	
AEGON UFK Alfa ETF 7 (PLN)	2020-01-21	11.6 PLN	0.00%	0.09%	0.61%	0.61%	3.48%	-0.43%	9.32%	
AEGON UFK Alfa ETF 8 (PLN)	2020-01-21	11.4 PLN	0.00%	-0.18%	0.09%	0.09%	2.25%	0.44%	8.40%	

Rysunek: Dane funduszy

```
public static double[][][] ReadData(String file, int sheet_count, int criteria_count, int funds_count) throws Exception
{
    double[][][] d = new double[sheet_count][criteria_count][funds_count];
    NumberCell number;
    Workbook port = Workbook.getWorkbook(new File(file)); // podpięcie do arkusza z portfelem
    Sheet portfel;
    for (int i = 0 ; i < sheet_count ; i++)
    {
      portfel = port.getSheet(i);
      for (int j = 1 ; j <= criteria_count ; j ++)
      {
          for (int k = 1 ; k <= funds_count ; k++)
          {
              number = (NumberCell) portfel.getCell(j,k);
              d[i][j-1][k-1] = number.getValue();
          }
      }
    }
    return d;
}
```

Rysunek: Działanie na arkuszu

Apache Pdfbox

- obsługa plików pdf;
- łączenie i dzielenie plików;
- pobieranie tekstu z pliku;
- tworzenie plików pdf;
- podpis cyfrowy plików.

```
public static void main(String[] args) throws Exception
{
 Connection connection = Jsoup.connect("https://www.sts.pl/pl/oferta/zaklady-bukmacherskie
 Document htmlDocument = connection.get();
 System.out.println(htmlDocument.body());
 URL url = new URL("https://www.sts.pl/userdata/files/pdf/oferta-jutro-www.pdf");
 InputStream in = url.openStream();
 Files.copy(in, Paths.get("STS.pdf"), StandardCopyOption.REPLACE_EXISTING);
 in.close();
```

```
PDDocument document = null;
document = PDDocument.load(new File("STS.pdf"));
document.getClass();
```

Rysunek: Pobieranie plików i obsługa PDF

```
if( !document.isEncrypted() )
{
    PDFTextStripperByArea stripper = new PDFTextStripperByArea();
    stripper.setSortByPosition( true );
    PDFTextStripper Tstripper = new PDFTextStripper();
    String st = Tstripper.getText(document);
    String[] temp = st.split("\n");
```

Rysunek: Pobranie tekstu z PDF

Zagadnienia na test

- pętle i instrukcje warunkowe;
- łańcuchy, metody związane z łańcuchami;
- łańcuch jako tablica;
- tablice 1D i 2D;
- tablice nieregularne;
- klasy: akcesory, konstruktory (domyślny, kopiujący, z parametrami);
- pliki tekstowe, obiektowe, RAF.

Dziękuję za uwagę