Przysłanianie metod

1. Przypomnienie przysłaniania konstruktorów
2. Przysłanianie metod

Trzy metody o tej samej nazwie!!!

```
public void empty(){
    System.out.println("Empty methos");
}

public void empty(String text){
    System.out.println("Metoda przesłonięta: text = "+text);
}

public void empty(String text, int ilosc){
    System.out.println("Kolejne przesłonięcie: text = "+text+"
ilosc = "+ilosc);
}
```

Jak to działa?

Wykona się ta metoda, która będzie miała wpisane odpowiednie parametry (lub nic nie będzie miała wpisane).

empty(”TAK”); - wykona się druga metoda bo przyjmuje ona dane typu tekstowego.

Przypomnienie metod zwracających wartość

```
//Metoda zwraca podwojoną wartość z wprowadzonej liczby
public BigDecimal dodawanie(BigDecimal l1){
    return l1.add(l1);
}
//Metoda dodaje dwie wielkie liczby i wzraca ich sumę
public BigDecimal dodawanie(BigDecimal l1,BigDecimal l2){
    return l1.add(l1);
}
```

W poniższych programach używamy zmiennej BigDecimal do wprowadzania liczb do metod. Oto jak tworzymy i wykonujemy działania na tej zmiennej

```
// Create two new BigDecimals
BigDecimal BigDec1 = new BigDecimal( val: "20");
BigDecimal BigDec2 = new BigDecimal( val: "5");
// Addition of two BigDecimals
BigDec1 = BigDec1.add(BigDec2);
System.out.println("Addition = " + BigDec1);
// Multiplication of two BigDecimals
BigDec1 = new BigDecimal( val: "20");
BigDec1 = BigDec1.multiply(BigDec2);
System.out.println("Multiplication = " + BigDec1);
// Subtraction of two BigDecimals
BigDec1 = new BigDecimal( val: "20");
BigDec1 = BigDec1.subtract(BigDec2);
System.out.println("Subtraction = " + BigDec1);
// Division of two BigDecimals
BigDec1 = new BigDecimal( val: "20");
BigDec1 = BigDec1.divide(BigDec2);
System.out.println("Division = " + BigDec1);
```

WYNIK DZIAŁANIA PROGRAMU

```
Addition = 25
Multiplication = 100
Subtraction = 15
Division = 4
```

DZIELENIE BigDecimal

Jeśli dzielimy niektóre liczby, to wynik dzielenie jest liczbą, która się nie kończy (np.: 1:3 = 0.3333….). Wtedy należy użyć w dzieleniu dwóch dodatkowych parametrów:

```
BigDec1 = BigDec1.divide(BigDec2, scale: 4, RoundingMode.valueOf(2));
```

WYNIK:

```
Division = 33.3334
```

scale – oznacza ilość miejsc po przecinku

RoundingMode.valueOf(2) – zaokrąglenie w górę
RoundingMode.valueOf(1) – zaokrąglenie w dół

Pierwiastek kwadratowy - BigDecimal:

```
//Ustawiamy precyzję wyniku
MathContext mc = new MathContext( setPrecision: 10);
BigDec1 = new BigDecimal( val: "4");
//obliczenie pierwiastka kwadratowego z dokładnością do 10 miejsc po przecinku
BigDec1 = BigDec1.sqrt(mc);
System.out.println(BigDec1);
//Pierwiastek z 2
BigDec1 = new BigDecimal( val: "2");
System.out.println(BigDec1.sqrt(mc));
```

Porównywanie BigDecimal

```
int res = bg1.compareTo(bg2); // compare bg1 with bg2
```

Jeśli res = -1 to bg1 jest mniejsze niż bg2

Jeśli res = 0 to bg1 = bg2

Jeśli res = 1 to bg1 jest większa niż bg2

bg1.compareTo(bg2); - wynikiem działania jest liczba -1, 0 lub 1

1. Napisz program obliczający pole trójkąta. W programie tym stwórz nową klasę o nazwie **AreaOfTheTriangle** w której umieścisz metody (przesłanianie metod czyli jedna metoda o tej samej nazwie) obliczające pole trójkąta. Użyj następujących wzorów (na celującą wzór z sinusami):

$$P_{\Delta ABC} = \frac{1}{2} \cdot a \cdot h_a \qquad P_{\Delta ABC} = \frac{abc}{4R} \qquad P_{\Delta ABC} = 2R^2 \cdot \sin\alpha \cdot \sin\beta \cdot \sin\gamma$$

$$P_{\Delta ABC} = rp \qquad P_{\Delta ABC} = \sqrt{p(p-a)(p-b)(p-c)}$$

2. Napisz program zapisujący do pliku dane: wzrost, waga, wiek (lub wzrost, waga lub tylko wzrost) w formacie **wzrost;waga;wiek**. Program ma też odczytywać dane z pliku, przy czym wyświetlać ma w następujący sposób:
   Wzrost: 168
   Waga: nie podano //jeśli brakuje wagi
   Wiek: nie podano //jeśli brakuje wieku