

Przesłanianie metod i pól

Wpisany przez Administrator

Z tej lekcji dowiesz się:

-jaki wpływ na dziedziczenie ma przeciążanie metod

-co to jest przysłanianie pól i metod

nauczysz się:

-przysłaniać pola i metody

Co powinieneś już wiedzieć?

-co to jest [dziedziczenie](#)

-co to jest [przeciążanie](#) metod

Wszystkie niezbędne wiadomości oraz wyjaśnienia potrzebne do zrozumienia tej lekcji znajdują się w poprzednich lekcjach w tym kursie.

Przypomnijmy z [poprzedniej lekcji](#): **przeciążanie metod** to używanie dwóch lub więcej metod

o takiej samej nazwie lecz z różnymi parametrami.

Przypomnijmy sobie, na rozgrzewkę kilka umiejętności dot. przeciążania metod. Spróbuj poniższe zadanie zrobić sam:

Zad. 1: Napisz program, który:

-definiuje klasę punkt, zawierającą

*pole typu double o nazwie *wspX*;

*pole typu double o nazwie *wspY*;

*metodę *ustaw_wspolrzedne*, która przyjmuje dwa argumenty typu double: *nX* oraz *nY* i podstawia te wartości odpowiednio w pola *wspX* oraz *wspY*

-definiuje klasę *punkt3d*, zawierającą:

*pole typu double o nazwie *wspZ*;

*metodę *ustaw_wspolrzedne*, która przyjmuje trzy argumenty typu double: *nX*, *nY* oraz *nZ* i podstawia te wartości odpowiednio w pola *wspX*

wspY

oraz

wspZ

Stwórz obiekt *p1* klasy *punkt* oraz obiekt *p2* klasy *pubkt3d*.

Wywołaj metodę *ustaw_wspolrzedne* (*double nX*, *double nY*) dla *p1* z dowolnymi poprawnymi wartościami (np.: 10, 20).

Wywołaj metodę *ustaw_wspolrzedne* (*double nX*, *double nY*) dla *p2* z dowolnymi poprawnymi wartościami (np.: 10, 20).

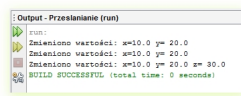
Wywołaj metodę *ustaw_wspolrzedne* (*double nX*, *double nY*, *double nZ*) dla *p2* z dowolnymi poprawnymi wartościami (np.: 10, 20, 30).

Spróbuj sam napisać ten program. Listing znajduje się na końcu instrukcji w dziale Rozwiązania.

Efekt uruchomienia programu:

Przesłanianie metod i pól

Wpisany przez Administrator



Co ustaliliśmy dzięki temu programowi?

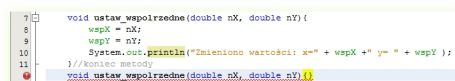
- 1) Możliwe jest przeciążanie metod w klasach bazowych i potomnych
- 2) Klasa potomna dziedziczy metody klasy bazowej
- 3) Klasa bazowa nie ma dostępu do metod klasy potomnej - co jest zupełnie logiczne. Jeśli nie wierzysz, to spróbuj wywołać metodę *ustaw_wspolrzedne* (*double nX, double nY, double nZ*) dla p2 z dowolnymi poprawnymi wartościami (np.: 10, 20, 30)

Przesłanianie metod

Czyli możliwe jest przeciążanie metod w dziedziczeniu. Pamiętajmy też, że warunkiem

koniecznym dla istnienia dwóch , metod o tej samej nazwie była różnica między nimi polegająca na zastosowaniu innej ilości lub innego typu parametrów. czyli nie można było zdefiniować dwóch metod o tej samej nazwie i tym samym zestawie parametrów. Przekonajmy się:

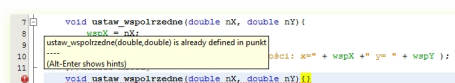
Zad. 2: Zmodyfikuj powyższy program: w klasie punkt dodaj jeszcze jedną metodę o nazwie *ustaw_wspolrzedne* , która przyjmuje dwa argumenty typu *double* :
nX
oraz
nY,
która nic nie robi.



```
void ustaw_wspolrzedne(double nX, double nY) {
    nX = nX;
    nY = nY;
    System.out.println("Zmieniono wartości: x=" + nX + " y=" + nY );
} //koniec metody
void ustaw_wspolrzedne(double nX, double nY) {}
```

Jak widać na powyższym obrazku, zdefiniowanie jeszcze raz metody o takiej samej nazwie i takich samych argumentach nie uda się: system wyświetla informację o błędzie (czerwone kółeczko z wykrzyknikiem):

Komunikat błędu:



```
void ustaw_wspolrzedne(double nX, double nY) {
    nX = nX;
    nY = nY;
    System.out.println("Zmieniono wartości: x=" + nX + " y=" + nY );
} //koniec metody
void ustaw_wspolrzedne(double nX, double nY) {}
```

Podana metoda jest już zdefiniowana w klasie punkt.

Teraz, idąc tym samym tokiem rozumowania, zdefiniowanie tej klasy w klasie *punkt3d*, która dziedziczy z klasy punkt powinno zakończyć się tym samym błędem - przecież dziedziczy ona z klasy punkt, więc będzie "miała" już odziedziczoną metodę

ustaw_wspolrzedne

, która przyjmuje dwa argumenty typu

double

:

nX

oraz

nY,

właśnie po klasie punkt. Sprawdźmy jednak, czy mamy rację:

Zad. 3: Zmodyfikuj powyższy program. Dodaj do klasy *punkt3d* metodę *ustaw_wspolrzedne*, która przyjmuje dwa argumenty typu

double

:

nX

oraz

nY.

(oczywiście usuń powodującą błąd metodę, którą dodaliśmy w zadaniu 2.)

Przesłanianie metod i pól

Wpisany przez Administrator

```
1 package przeslanianie;
2 public class Przeslanianie {
3     public static void main(String[] args) {
4
5         class punkt {
6             double wspX, wspY;
7             void ustaw_wspolrzedne(double nX, double nY) {
8                 wspX = nX;
9                 wspY = nY;
10                System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
11            }
12        }
13        //koniec klasy punkt
14        class punkt3d extends punkt {
15            double wspZ;
16            void ustaw_wspolrzedne(double nX, double nY, double nZ) {
17                wspX = nX;
18                wspY = nY;
19                wspZ = nZ;
20                System.out.println("Zmieniono wartosci: x=");
21                System.out.println(wspX + " y=" + wspY + " z=" + wspZ);
22            }
23        }
24        //koniec klasy punkt3d
25        punkt p1 = new punkt();
26        punkt3d p2 = new punkt3d();
27        p1.ustaw_wspolrzedne(10, 20); //metoda w klasie bazowej
28        p2.ustaw_wspolrzedne(10, 20); //metoda odziedziczona z klasy bazowej
29        p2.ustaw_wspolrzedne(10, 20, 30); //metoda w klasie potomnej
30    }
31 }
```

```
1 package przeslanianie;
2 public class Przeslanianie {
3     public static void main(String[] args) {
4
5         class punkt {
6             double wspX, wspY;
7             void ustaw_wspolrzedne(double nX, double nY) {
8                 wspX = nX;
9                 wspY = nY;
10                System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
11                System.out.println("Jestem czescia klasy punkt");
12            }
13        }
14        //koniec klasy punkt
15        class punkt3d extends punkt {
16            double wspZ;
17            void ustaw_wspolrzedne(double nX, double nY, double nZ) {
18                wspX = nX;
19                wspY = nY;
20                wspZ = nZ;
21                System.out.println("Zmieniono wartosci: x=");
22                System.out.println(wspX + " y=" + wspY + " z=" + wspZ);
23            }
24            void ustaw_wspolrzedne(double nX, double nY) {
25                wspX = nX;
26                wspY = nY;
27                System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
28                System.out.println("Jestem czescia klasy punkt3d");
29            }
30        }
31        //koniec klasy punkt3d
32        punkt p1 = new punkt();
33        punkt3d p2 = new punkt3d();
34        p1.ustaw_wspolrzedne(10, 20); //metoda w klasie bazowej
35        p2.ustaw_wspolrzedne(10, 20); //metoda odziedziczona z klasy bazowej
36        p2.ustaw_wspolrzedne(10, 20, 30); //metoda w klasie potomnej
37    }
38 }
```

Output - Przeslanianie (run)

```
run:
Zmieniono wartosci: x=10.0 y= 20.0
Jestem czescia klasy punkt
Zmieniono wartosci: x=10.0 y= 20.0
Jestem czescia klasy punkt3d
Zmieniono wartosci: x=10.0 y= 20.0 z= 30.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
1 package przeslanianie;
2 public class Przeslanianie {
3     public static void main(String[] args) {
4
5         class punkt {
6             double wspX, wspY;
7             void ustaw_wspolrzedne(double nX, double nY) {
8                 wspX = nX;
9                 wspY = nY;
10                System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
11                System.out.println("Jestem czescia klasy punkt");
12            }
13        }
14        //koniec klasy punkt
15        class punkt3d extends punkt {
16            double wspZ;
17            void ustaw_wspolrzedne(double nX, double nY, double nZ) {
18                wspX = nX;
19                wspY = nY;
20                wspZ = nZ;
21                System.out.println("Zmieniono wartosci: x=");
22                System.out.println(wspX + " y=" + wspY + " z=" + wspZ);
23            }
24            void ustaw_wspolrzedne(double nX, double nY) {
25                wspX = nX;
26                wspY = nY;
27                System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
28                System.out.println("Jestem czescia klasy punkt3d");
29            }
30            void ustaw_wspolrzedne(double nX, double nY) {
31                super.ustaw_wspolrzedne(nX, nY); //odwołanie do bazowej metody
32            }
33        }
34        //koniec klasy punkt3d
35        punkt p1 = new punkt();
36        punkt3d p2 = new punkt3d();
37        p1.ustaw_wspolrzedne(10, 20); //metoda w klasie bazowej
38        p2.ustaw_wspolrzedne(10, 20); //metoda odziedziczona z klasy bazowej
39        p2.ustaw_wspolrzedne(10, 20, 30); //metoda odziedziczona z klasy bazowej
40        p2.ustaw_wspolrzedne(10, 20, 30); //metoda w klasie potomnej
41    }
42 }
```

```
1 punkt p1 = new punkt();
2 punkt3d p2 = new punkt3d();
3 p1.ustaw_wspolrzedne(10, 20); //metoda w klasie bazowej
4 p2.ustaw_wspolrzedne(10, 20); //metoda odziedziczona z klasy bazowej
5 p2.ustaw_wspolrzedne(10, 20, 30); //metoda odziedziczona z klasy bazowej
6 p2.ustaw_wspolrzedne(10, 20, 30); //metoda w klasie potomnej
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Output - Przeslanianie (run)

```
run:
Zmieniono wartosci: x=10.0 y= 20.0
Jestem czescia klasy punkt
Zmieniono wartosci: x=10.0 y= 20.0
Jestem czescia klasy punkt3d
Zmieniono wartosci: x=10.0 y= 20.0
Jestem czescia klasy punkt
Zmieniono wartosci: x=10.0 y= 20.0 z= 30.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Przebieg, który prowadzi do wywołania metody w klasie bazowej, a nie w klasie potomnej, jest następujący:

Przesłanianie metod i pól

Wpisany przez Administrator

```
35 punkt p1 = new punkt();
36 punkt3d p2 = new punkt3d();
37 p1.ustaw_wspolrzedne(10, 20); //metoda w klasie bazowej
38 p2.ustaw_wspolrzedne(10, 20); //metoda odziedziczona z klasy bazowej
39 p2.ustaw_wspolrzedne2(10, 20, 30); //metoda w klasie potomnej
40 }
41
42 Output - Przesłanianie (run)
43 run:
44 Zmieniono wartosci: x=10.0 y= 20.0
45 Jestem częścią klasy punkt
46 Zmieniono wartosci: x=10.0 y= 20.0
47 Jestem częścią klasy punkt3d
48 Zmieniono wartosci: x=10.0 y= 20.0 z= 30.0
49 BUILD SUCCESSFUL (total time: 0 seconds)
```

```
1 package przeslanianie;
2 public class Przeslanianie {
3     public static void main(String[] args) {
4     }
5 }
6 class punkt {
7     double wspX, wspY;
8     void ustaw_wspolrzedne(double nX, double nY) {
9         wspX = nX;
10        wspY = nY;
11        System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
12        System.out.println("Jestem częścią klasy punkt");
13    } //koniec metody
14 } //koniec klasy punkt
15 class punkt3d extends punkt {
16     double wspZ;
17     void ustaw_wspolrzedne(double nX, double nY, double nZ) {
18         wspX = nX;
19         wspY = nY;
20         wspZ = nZ;
21        System.out.print("Zmieniono wartosci: x=");
22        System.out.println(wspX + " y=" + wspY + " z=" + wspZ);
23    } //koniec metody
24    void ustaw_wspolrzedne2(double nX, double nY) {
25        wspX = nX;
26        wspY = nY;
27        System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
28        System.out.println("Jestem częścią klasy punkt3d");
29    }
30    void ustaw_wspolrzedne3(double nX, double nY) {
31        super.ustaw_wspolrzedne(nX, nY); //odwołanie do bazowej metody
32    }
33 } //koniec klasy punkt3d
34 punkt p1 = new punkt();
35 punkt3d p2 = new punkt3d();
36 p1.ustaw_wspolrzedne(10, 20); //metoda w klasie bazowej
37 p2.ustaw_wspolrzedne(10, 20); //metoda odziedziczona z klasy bazowej
38 p2.ustaw_wspolrzedne2(10, 20); //metoda odziedziczona z klasy bazowej
39 p2.ustaw_wspolrzedne3(10, 20, 30); //metoda w klasie potomnej
40 }
```

```
14 class punkt3d extends punkt {
15     double wspX, wspY, wspZ;
16     void ustaw_wspolrzedne(double nX, double nY, double nZ) {
17         wspX = nX;
18         wspY = nY;
19         wspZ = nZ;
20        System.out.print("Zmieniono wartosci: x=");
21        System.out.println(wspX + " y=" + wspY + " z=" + wspZ);
22    } //koniec metody
23    void ustaw_wspolrzedne2(double nX, double nY) {
24        wspX = nX;
25        wspY = nY;
26        System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
27    }
28    void ustaw_wspolrzedne3(double nX, double nY) {
29        super.ustaw_wspolrzedne(nX, nY); //odwołanie do bazowej metody
30    }
31    void ustaw_wspolrzedne3(double nX, double nY) {
32        super.wspX = nX;
33        super.wspY = nY;
34    }
35    void wywiewi_wspolrzedne() {
36        System.out.println("wspX odziedziczona = " + super.wspX); //pole klasy punkt
37        System.out.println("wspY odziedziczona = " + super.wspY); //pole klasy punkt
38        System.out.println("wspX klasy potomnej = " + this.wspX); //pole klasy punkt3d
39        System.out.println("wspY klasy potomnej = " + this.wspY); //pole klasy punkt3d
40    }
41 } //koniec klasy punkt3d
42 punkt3d p2 = new punkt3d();
43 p2.ustaw_wspolrzedne(30, 40); //ustawianie wartosci pól pochodnych
44 p2.ustaw_wspolrzedne3(50, 60); //ustawianie wartosci pól pochodnych
45 p2.wywiewi_wspolrzedne();
46 }
47 }
```

Oto wynik uruchomienia programu:

```
28 void ustaw_wspolrzedne2(double nX, double nY) {
29     super.ustaw_wspolrzedne(nX, nY); //odwołanie do bazowej metody
30 }
31 void ustaw_wspolrzedne3(double nX, double nY) {
32     super.wspX = nX;
33     super.wspY = nY;
34 }
35 void wywiewi_wspolrzedne() {
36     System.out.println("wspX odziedziczona = " + super.wspX); //pole klasy punkt
37     System.out.println("wspY odziedziczona = " + super.wspY); //pole klasy punkt
38     System.out.println("wspX klasy potomnej = " + this.wspX); //pole klasy punkt3d
39     System.out.println("wspY klasy potomnej = " + this.wspY); //pole klasy punkt3d
40 }
41 } //koniec klasy punkt3d
42 punkt3d p2 = new punkt3d();
43 p2.ustaw_wspolrzedne(30, 40); //ustawianie wartosci pól pochodnych
44 p2.ustaw_wspolrzedne3(50, 60); //ustawianie wartosci pól pochodnych
45 p2.wywiewi_wspolrzedne();
46 }
47 }
48
49 Output - Przesłanianie (run)
50 run:
51 Zmieniono wartosci: x=30.0 y= 40.0
52 wspX odziedziczona = 30.0
53 wspY odziedziczona = 40.0
54 wspX klasy potomnej = 50.0
55 wspY klasy potomnej = 60.0
56 BUILD SUCCESSFUL (total time: 1 seconds)
```


Przesłanianie metod i pól

Wpisany przez Administrator

```
1 package przeslanianie;
2 public class Przeslanianie {
3     public static void main(String[] args) {
4
5     }
6     class punkt1 {
7         double wspX, wspY;
8         void ustaw_wspolrzedne(double nX, double nY) {
9             wspX = nX;
10            wspY = nY;
11            System.out.println("Zmieniono wartosci: x=" + wspX + " y=" + wspY);
12        }
13    }
14    class punkt3d extends punkt1 {
15        double wspZ;
16        void ustaw_wspolrzedne(double nX, double nY, double nZ) {
17            wspX = nX;
18            wspY = nY;
19            wspZ = nZ;
20            System.out.print("Zmieniono wartosci: x=");
21            System.out.println(wspX + " y=" + wspY + " z=" + wspZ);
22        }
23    }
24    punkt1 p1 = new punkt1();
25    punkt3d p2 = new punkt3d();
26    p1.ustaw_wspolrzedne(10, 20); //metoda w klasie bazowej
27    p2.ustaw_wspolrzedne(10, 20, 30); //metoda w klasie potomnej
28 }
29 }
```

Zad. 9.

```
1 package utrwalanie2;
2 public class Utrwalanie2 {
3     public static void main(String[] args) {
4         class kwadrat {
5             double bokA;
6             void ustaw_a (double nA) { //metoda bazowa
7                 bokA = nA;
8             }
9             kwadrat() { //konstruktor
10                ustaw_a(56);
11            }
12            void wyswietl_a() {
13                System.out.println("Wartość boku a: " + bokA);
14            }
15        }
16        class prostokat extends kwadrat {
17            double bokB;
18            void ustaw_b (double nB) {
19                bokB = nB;
20            }
21            prostokat() { //konstruktor
22                ustaw_b (65);
23            }
24            void wyswietl_b() { //metoda klasy pochodnej
25                System.out.println("Wartość boku b: " + bokB);
26            }
27        }
28        kwadrat kw1 = new kwadrat();
29        prostokat pr1 = new prostokat();
30        System.out.println("kwadrat:");
31        kw1.wyswietl_a();
32        System.out.println("prostokat:");
33        pr1.wyswietl_a();
34        pr1.wyswietl_b();
35    }
36 }
37 }
```

Output - utrwalanie2 (run)

```
run:
kwadrat:
Wartość boku a: 56.0
prostokat:
Wartość boku a: 56.0
Wartość boku b: 65.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Zad. 9.

```
1 package utrwalanie2;
2 public class Utrwalanie2 {
3     public static void main(String[] args) {
4         class kwadrat {
5             double bokA;
6             void ustaw_a (double nA) { //metoda bazowa
7                 bokA = nA;
8             }
9             kwadrat() { //konstruktor
10                ustaw_a(56);
11            }
12            void wyswietl_a() {
13                System.out.println("Wartość boku a: " + bokA);
14            }
15        }
16        class prostokat extends kwadrat {
17            double bokB;
18            void ustaw_b (double nB) {
19                bokB = nB;
20            }
21            void ustaw_a (double nA) { //metoda bazowa
22                bokA = nA;
23            }
24            void ustaw_a_bazowe (double nA) {
25                super.ustaw_a(nA);
26            }
27            prostokat() { //konstruktor
28                ustaw_b (65);
29            }
30            void wyswietl_b() { //metoda klasy pochodnej
31                System.out.println("Wartość boku b: " + bokB);
32            }
33        }
34        kwadrat kw1 = new kwadrat();
35        prostokat pr1 = new prostokat();
36        System.out.println("kwadrat:");
37        kw1.wyswietl_a();
38        System.out.println("prostokat:");
39        pr1.wyswietl_a();
40        pr1.wyswietl_b();
41        pr1.ustaw_a_bazowe(12);
42        pr1.wyswietl_a();
43    }
44 }
45 }
```

Zad. 10.:

```
1 package utrwalanie2;
2 public class Utrwalanie2 {
3     public static void main(String[] args) {
4         class Kwadrat {
5             double bokA;
6             void ustaw_a (double nA) { //metoda bazowa
7                 bokA = nA;
8             }
9             Kwadrat() { //konstruktor
10                ustaw_a(56);
11            }
12            void wyswietl_a() {
13                System.out.println("Wartość boku a: " + bokA);
14            }
15        }
16
17        class prostokat extends Kwadrat {
18            double bokA, bokB;
19            void ustaw_b (double nB) {
20                bokB = nB;
21            }
22            void ustaw_a (double nA) { //metoda bazowa
23                bokA = nA;
24            }
25            void ustaw_a_bazowe (double nA) {
26                super.ustaw_a(nA);
27            }
28            void ustaw_a_bazowe2 (double nA) {
29                super.bokA = nA;
30            }
31            prostokat() { //konstruktor
32                ustaw_b (65);
33            }
34            void wyswietl_b() { //metoda klasy pochodnej
35                System.out.println("Wartość boku b: " + bokB);
36            }
37        }
38
39        Kwadrat kw1 = new Kwadrat();
40        prostokat pri = new prostokat();
41        System.out.println("Kwadrat:");
42        kw1.wyswietl_a();
43        System.out.println("prostokat:");
44        pri.wyswietl_a();
45        pri.wyswietl_b();
46        pri.ustaw_a_bazowe(12);
47        pri.wyswietl_a();
48    }
49 }
```

Dziękuję za uwagę.