



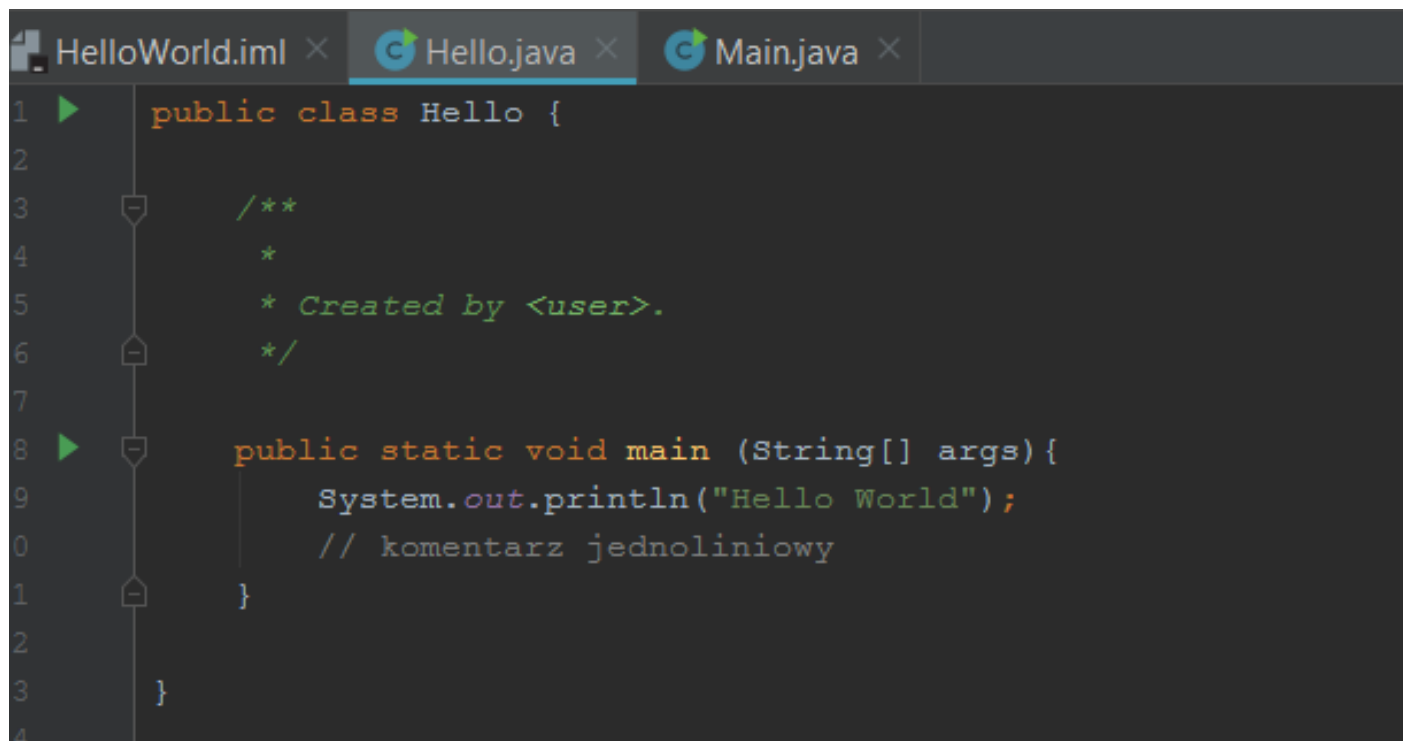
Wyższa Szkoła Bankowa
Gdańsk Gdynia

Analiza danych w języku Java I

Obiekty, metody, komentarze

Anna Pakeizer

Komentarze

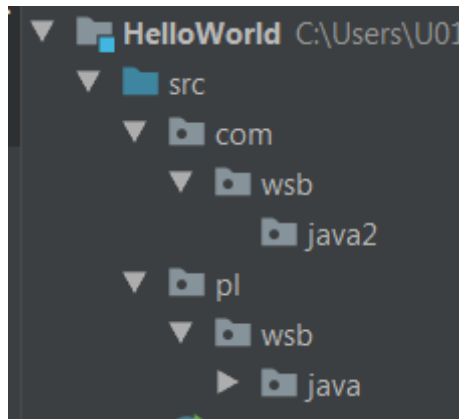


```
1 public class Hello {
2
3     /**
4      *
5      * Created by <user>.
6      */
7
8     public static void main (String[] args){
9         System.out.println("Hello World");
10        // komentarz jednoliniowy
11    }
12
13 }
```

Pakiety

```
package pl.wsb.java;
```

- Pakiety służą do grupowania klas;
- package – słowo kluczowe; informuje kompilator w jakim pakiecie znajduje się plik;
- pl.wsb.java – właściwa nazwa pakietu.
- Każdy człon pakietu odpowiada katalogowi na dysku.



Klasa

1. Co definiuje słowo **class**?

Nowy typ.

2. Co opisuje klasa?

Zbiór obiektów o tej samej charakterystyce, tj. określa jakie obiekt może mieć pola i metody.

3. Co to jest klasa?

Klasa określa, jakie obiekty mogą powstać w oparciu o definicję klasy.

4. Ile obiektów można stworzyć, gdy klasa jest już stworzona?

Dowolną liczbę.

Klasa

```
public class Hello {}
```

- *public* – modyfikator dostępu (*public* oznacza że jest widoczna dla innych klas)
- *class* – słowo kluczowe informujące kompilator, że ma do czynienia z definicją klasy
- *Hello* – nazwa klasy
- *{ }* - blok grupujący instrukcje – definicja klasy.

Modyfikatory dostępu

- *public* – oznacza, że jest widoczna dla innych klas;
- *private* – oznacza, że dany element dostępny jest wyłącznie z wnętrza danej klasy.

Modyfikator	Klasa	Pakiet	Podklasa	Inni	Poprawny dla klas
<code>public</code>	tak	tak	tak	tak	tak
<code>protected</code>	tak	tak	tak	nie	nie
brak modyfikatora	tak	tak	nie	nie	tak
<code>private</code>	tak	nie	nie	nie	nie

Klasa c.d.

```
public class Zawodnik {  
  
    int wiek;  
    int numerStartowy;  
  
    public Zawodnik() {  
    }  
  
    public Zawodnik(int wiek, int nrStartowy) {  
        this.wiek = wiek;  
        numerStartowy = nrStartowy;  
    }  
}
```

- atrybuty klasy:
- konstruktory:

```
int wiek;
```

```
int numerStartowy;
```

```
public Zawodnik() {  
}  
  
public Zawodnik(int wiek, int nrStartowy) {  
    this.wiek = wiek;  
    numerStartowy = nrStartowy;  
}
```

Konstruktory

```
public Zawodnik() {  
}  
  
public Zawodnik(int wiek, int nrStartowy) {  
    this.wiek = wiek;  
    numerStartowy = nrStartowy;  
}
```

- Konstruktor służy do tworzenia nowych instancji klasy. Każda klasa musi mieć konstruktor. Konstruktory to specjalne metody, które inicjalizują instancje klas.
- Konstruktory wywołuje się dokładnie tak samo jak inne metody, dodatkowo używamy słowa kluczowego *new*

```
Zawodnik z = new Zawodnik();  
Zawodnik x = new Zawodnik(35,1);
```

- Kompilator tworzy domyślny konstruktor automatycznie jeśli programista nie zdefiniuje żadnego innego konstruktora.

Konstruktory - słowo kluczowe *this*

```
public class Zawodnik {  
  
    int wiek;  
    int numerStartowy;  
  
    public Zawodnik() {  
    }  
  
    public Zawodnik(int wiek, int nrStartowy) {  
        this.wiek = wiek;  
        numerStartowy = nrStartowy;  
    }  
}
```

W pierwszym przypadku w konstruktorze zostało użyte słowo kluczowe *this*. Było ono potrzebne aby kompilator był w stanie odróżnić parametr *wiek* od atrybutu klasy *wiek*. W drugim przypadku słowo to nie było konieczne ponieważ w tym kontekście od razu wiadomo czym jest *numerStartowy*– jest atrybutem klasy. Oba konstruktory robią dokładnie to samo, przypisują wartość parametru to atrybutu nowej instancji.

Metoda

- Metoda to zestaw instrukcji, który opcjonalnie może zwracać jakąś wartość (worek grupujący zestaw instrukcji).

```
<typ zwracany> <nazwa metody>(<opcjonalna lista argumentów>) {  
    <ciało metody>  
}
```

```
boolean isBig(int someNumber) {  
    return someNumber > 100;  
}
```

- Wywołanie metody:

```
boolean someVariable = isBig(10);
```

```
printSomething();
```

Metoda - wartość zwracana

- Metoda może zwracać jakąś wartość. Do zwrócenia tej wartości służy słowo *return*

```
String getName(){  
    return "Jacek";  
}
```

- Metoda może nie zwracać żadnej wartości, używamy wtedy słowa kluczowego *void* do określenia typu zwracanej wartości

```
void printSomethig() {  
    System.out.println("Something");  
}
```

- Metody mogą przyjmować wiele argumentów. W języku Java nie jest możliwe zwracanie wielu wartości. Metoda może nie zwrócić nic (*void*) lub pojedynczą wartość.

Auto activation triggers

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ._

