



Język JAVA podstawy programowania

Jacek Rumiński

wykład 1, część 3

Plan wykładu:

1. Krótka historia Javy
2. Jak przygotować sobie środowisko programistyczne
3. Opis środowiska JDK
4. Tworzenie programu „krok po kroku”

Procedura „krok po kroku”:

1. Napisanie z pomocą dowolnego edytora tekstu (lub IDE) kodu źródłowego programu zawierającego klasę publiczną o nazwie takiej samej (dokładnie takiej samej z uwzględnieniem wielkości znaków) jak docelowa nazwa programu np. **RycerzJedi**.
2. Nagranie kodu źródłowego jako pliku o danej nazwie z rozszerzeniem „.java”, np. **RycerzJedi.java**
3. Kompilacja kodu źródłowego zawartego w pliku z rozszerzeniem .java do pliku docelowego o rozszerzeniu .class zawierającego Beta-kod np. **c:\> javac -g RycerzJedi.java**
4. Uruchomienie w środowisku maszyny wirtualnej kodu pośredniego (uniwersalnego), np. **c:\> java RycerzJedi**

Ale – Co to jest „klasa”

Język Java jest językiem obiektowym. Oznacza to, że tworzenie programów związane jest z pewnym sposobem modelowania tego, co nas otacza oraz tego, co sobie wyobrażamy. Programowanie obiektowe wymusza zastosowanie określonej formy kodu źródłowego oraz umożliwia wykorzystanie szeregu ciekawych cech, które poznamy później.

Klasa oznacza pewien typ (rodzaj, kategorię). Posłużmy się przykładem ryb. Już samo pojęcie „ryba” oznacza pewien „rodzaj” (pojęcie ogólne). Czy istnieje ryba jako ryba? Nie. Istnieje konkretna ryba, którą złapaliśmy na wędkę : ma określone cechy: kolor łusek, ilość płetw oraz określone zachowanie (np. wije się – węgorz?). Co więcej, złapana ryba istnieje konkretnie (w danym miejscu i czasie – na haczyku, teraz).

Zatem: **RYBA** = **KLASA** (zestaw cech i zachowania jakie mają ryby)
ZŁAPANA RYBA = **OBIEKT** klasy **RYBA** (konkretne wystąpienie typu).

Ale – co to jest „klasa”

Opiszmy rybę:

klasa RYBA:

cechy:

kolor oczu;
kolor łusek;
liczba płetw,
płeć;
czy_drapieżna;

zachowanie:

pływa;
żeruje;

obiekt klasy RYBA

=niebieskie
=pomarańczowe
=5
=samica
=nie

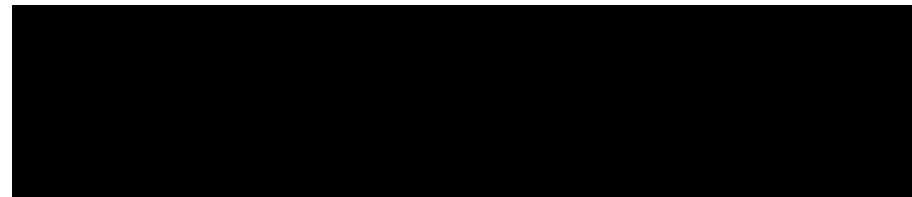
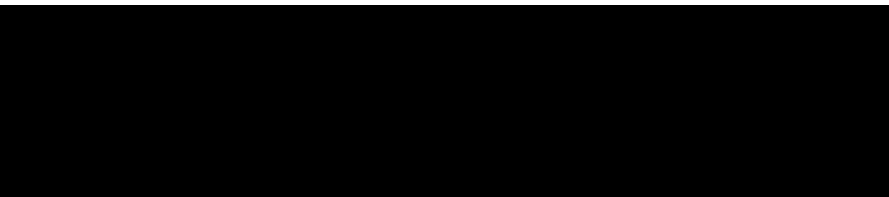
pływa
żeruje

Ale – co to jest „obiekt”?

Jeśli utworzyliśmy (zaprojektowaliśmy) daną klasę (RYBA) to możemy utworzyć wiele obiektów danej klasy z różnymi cechami.

OBIEKT – wystąpienie (instancja) KLASY

Obiekty w świecie rzeczywistym istnieją konkretnie (w czasie i miejscu). Obiekty w świecie komputerowym będą „żyły” również w określonym czasie i miejscu - w pamięci (operacyjnej RAM, lub utrwalone w pamięci nieulotnej).



Możemy utworzyć wiele obiektów danej klasy !!!

Ale – co to jest „obiekt”?

```
Okoń1 = nowy obiekt (klasy RYBA);  
Okoń1(kolor oczu)=czerwony;  
Okoń1(płeć)=samiec;  
...
```

Konstruktor –
„tworzy” obiekt

Ustawienie wartości
cech danej (konkretnej)
ryby

Okoń1 = nowy obiekt (klasy RYBA);

Płotka1 = nowy obiekt (klasy RYBA);

Sandacz1= nowy obiekt (klasy RYBA);

Pamięć RAM

adres 2354635261

adres 4354635264

adres 5354459863

Nazwy: „Okoń1”, „Sandacz1” to „uchwyty” wskazujące na pamięć (o jaki obiekt nam chodzi).

Skrót myślowy: obiekt „Okoń1” – znaczenie: obiekt pod adresem...

Ale – co to jest „obiekt”?

Tożsamość i unikalność obiektu określana jest nie przez wartości cech ale przez instancję (tj. konkretne wystąpienie w określonym czasie pod danym adresem pamięci).

Dwa obiekty mogą mieć TE SAME wartości wszystkich cech, ale są różne (NOT EQUAL).

Modelowanie obiektowe posiada szereg innych cech (np. dziedziczenie klas – typ KOBIECIA jest podtypem – dziedziczy po typie CZŁOWIEK), które omówimy później.

Teraz musimy przedstawić zasady zapisu klas i obiektów w języku JAVA. Każdy język obiektowy będzie miał własną metodę (składnię) zapisu definiującego klasy i obiekty. Komputery są bardzo konkretne – konkretny (i dokładny co do znaku) musi być również zapis kodu źródłowego definiujący klasę i obiekt.

Jak zamodelować klasę w JAVIE?

Każdy język programowania ma zestaw słów (zwykle w j. angielskim) zarezerwowanych do zapisu treści (instrukcji, rozkazów) programu. Do zamodelowania pierwszej klasy wykorzystamy następujące słowa kluczowe:

class - początek definiowania klasy,

public - specyfikator oznaczający dostępność (*public* – wszystkie obiekty mogą korzystać),

int - typ danych liczb całkowitych (*integer*),

void – typ danych oznaczający „nic”,

boolean – logiczny typ danych (wartość *true* lub *false*),

String – typ danych znakowych (który też jest klasą, stąd pisany z wielkiej litery).

Ponadto zastosujemy oznaczenie nawiasów klamrowych do określania (grupowania) bloku kodu – { }

Kod programu: Ryba.java

```
public class Ryba {
    //cechy
    String kolor_lusek;
    int liczba_pletw;
    boolean drapiezna;

    //konstruktor
    public Ryba (String kl, int lp, boolean d){
        kolor_lusek=kl;
        liczba_pletw=lp;
        drapiezna=d;
    }//koniec Ryba()

    //działanie – funkcje, metody
    public void plywa(){
        System.out.println("Tak plywam: plum plum plum! ");
    }//koniec plywa()

} //koniec klasy Ryba
```



DEMO



Wszystkie przykładowe pliki programów załączono na stronie kursu (SZKOLENIA/JAVA)
<http://uno.biomed.gda.pl>

Jak zamodelować klasę w JAVIE?

W przykładzie konstruktor to funkcja trzech zmiennych. Można go wywołać podstawiając wartości danego typu – tworzymy obiekt z określonymi wartościami cech.

Zamiast używać różnych nazw zmiennych (np. nazwa pola: „kolor_lusek”, nazwa zmiennej „kl”) można wykorzystać kolejne słowo kluczowe **this** („w tym obiekcie”).

```
String kolor_lusek;  
int liczba_pletw;  
boolean drapiezna;  
//konstruktor  
public Ryba (String kolor_lusek, int liczba_pletw, boolean drapiezna){  
    this.kolor_lusek=kolor_lusek;  
    this.liczba_pletw=liczba_pletw;  
    this.drapiezna=drapiezna;  
} //koniec Ryba()
```

Czy to był program?

1. Kompilacja Ryba.java - OK.

2. Uruchomienie – NIE!!!

Co utworzyliśmy? Klasę, z której możemy skorzystać tak samo jak z wielu klas dostępnych w standardowej dystrybucji Javy.

Klasę można traktować jako najmniejszy element biblioteki, z której możemy wielokrotnie korzystać (ang. reuse).

Ale jak coś uruchomić!?

Zapraszam na wykład 2