

Materiał pomocniczy do kursu „Podstawy programowania”

Autor: Grzegorz Góralski

ggoralski.com

Klasy i obiekty cz I

Klasy, obiekty, podstawy używania obiektów

Obiekty

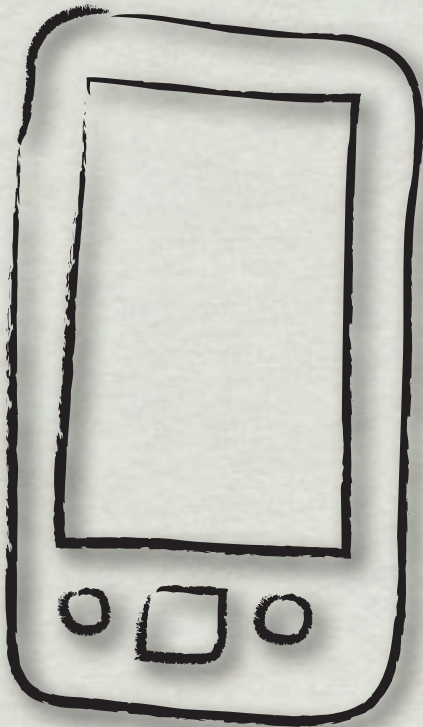
- * W codziennym życiu mamy do czynienia z różnego rodzaju obiektami, jak np. zwierzęta, samochody, czy telefony komórkowe
- * mają one swoje cechy, właściwości oraz mogą wykonywać (być użyte do wykonania) pewne czynności

WŁAŚCIWOŚCI/CECHY/DANE:

- * marka
- * model
- * numer
- * kontakty
- * kolor
- * ...

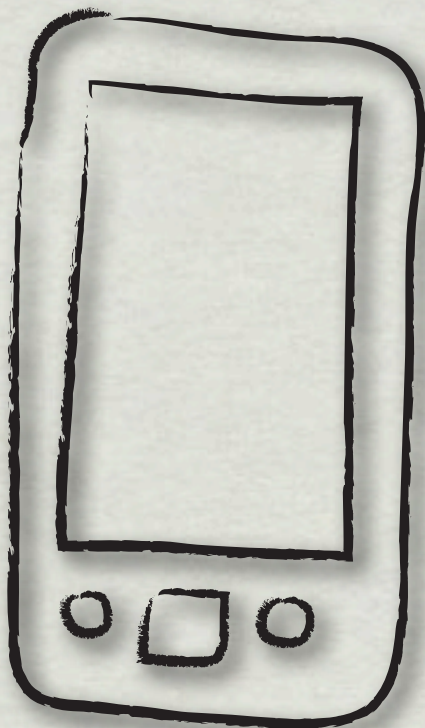
CZYNNOŚCI:

- * dzwonienie
- * odbieranie połączeń
- * wysyłanie SMS
- * odbieranie SMS
- * wyszukiwanie kontaktów
- * dodawanie kontaktów
- * ...



Obiekty

- * Przenosząc obraz rzeczywistych (lub wymyślonych) obiektów w świat programu, odwzorowujemy:
 - * właściwości/cechy/dane na **поля (dane, składowe)**
 - * czynności na **metody**



WŁAŚCIWOŚCI/CECHY/DANE:

- * marka
- * model
- * numer
- * kontakty
- * kolor
- * ...

CZYNNOŚCI:

- * dzwonenie
- * odbieranie połączeń
- * wysyłanie SMS
- * odbieranie SMS
- * wyszukiwanie kontaktów
- * dodawanie kontaktów
- * ...

POLA:

- * marka
- * model
- * numer
- * kontakty
- * kolor
- * ...

METODY:

- * zadzwon()
- * odbierzPołączenie()
- * wyslijSMS()
- * odbierzSMS
- * wyszukajKontakt()
- * dodajKontaktów()
- * ...

Tworzymy klasę

- * Pierwszym krokiem do utworzenia wirtualnego telefonu jest napisanie klasy, umieścimy go w pakiecie telefony:

```
public class Telefon {
```

```
    String marka;  
    String model;  
    int numer;  
    String kontakty;  
    String kolor;
```

POLA

```
    public void zadzwon(int nr){  
        System.out.println("Dzwonię z nr: "+numer);  
        System.out.println("Na nr: "+nr);  
    }  
    public void odbierzPolaczenie(int nr) {  
        System.out.println("Odbieram połączenie od nr: "+nr);  
    }  
    public void wyslijSMS(int nr, String tekst) {  
        System.out.println("Wysyłam SMS na nr: "+nr);  
        System.out.println("O treści: "+tekst);  
    }  
    public String odbierzSMS() {  
        String tekstSMS="Witaj telefonie!";  
        return tekstSMS;  
    }  
}
```

METODY

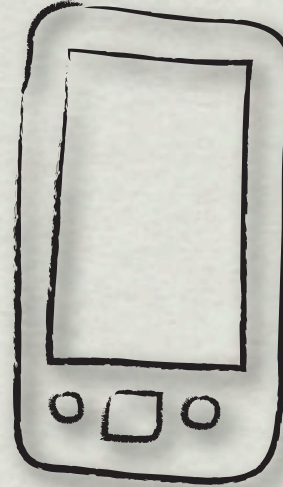
Klasy i obiekty

- * Klasa to jednak nie jest jeszcze „wirtualny telefon”
- * Na podstawie klasy tworzy się dopiero obiekty
- * Klasa jest więc „formą”, „planem” czy też „szablonem” na podstawie którego tworzy się obiekty
- * Obiekty posiadają konkretne właściwości i wykonują poszczególne czynności.

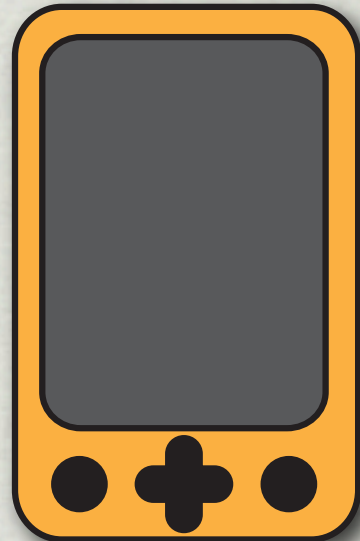
Klasy i obiekty

KLASA:

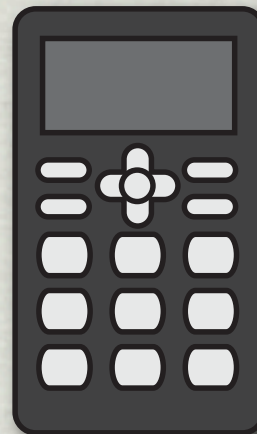
Telefon



Telefon1



Telefon2



Telefon3



OBIEKTY:

Tworzymy obiekty i używamy ich

- * Tworzenie obiektu:

```
Telefon tel1 = new Telefon();
```

- * Ustawianie wartości pola obiektu (bezpośrednie):

```
tel1.kolor="Niebieski";
```

- * Pobranie wartości pola obiektu (bezpośrednie):

```
tel1.kolor;
```

- * Wywołanie metody obiektu i pobranie zwracanej przez nią wartości:

```
String sms = tel1.odbierzSMS();
```

- * Wywołanie metody obiektu i przekazanie parametrów metodzie:

```
tel1.wyslijSMS(333444, "Cześć! To ja!");
```

Tworzymy obiekty i używamy ich

- * W tym samym pakiecie, tworzymy klasę z metodą main() o treści:

```
public class Telefony {
```

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

```
        Telefon tel1 = new Telefon();
```

```
        Telefon tel2 = new Telefon();
```

```
        tel1.marka="Motorola";
```

```
        tel1.numer=123456;
```

```
        tel1.kolor="Niebieski";
```

```
        tel2.marka="Nokia";
```

```
        tel2.numer=789000;
```

```
        tel2.kolor="Czarny";
```

```
        System.out.println("Marka telefonu 1: "+tel1.marka);
```

```
        System.out.println("Numer telefonu 1: "+tel1.numer);
```

```
        System.out.println("Kolor telefonu 1: "+tel1.kolor);
```

```
        System.out.println("Marka telefonu 2: "+tel2.marka);
```

```
        System.out.println("Numer telefonu 2: "+tel2.numer);
```

```
        System.out.println("Kolor telefonu 2: "+tel2.kolor);
```

```
        tel1.zadzwon(222333);
```

```
        tel2.wyslijSMS(333444, "Cześć! To ja!");
```

```
        String sms = tel2.odbierzSMS();
```

```
        System.out.println("Odebrałem SMS o treści: "+sms);
```

```
        tel1.wyslijSMS(tel2.numer, "Jaki ja?");
```

```
    }
```

```
}
```

Tworzymy dwa obiekty typu Telefon

Przypisujemy wartości wybranym składowym (polom) obu obiektów

Odwołujemy się do wybranych pól obu obiektów.

Wywołujemy wybrane metody obu obiektów.

Wynik działania programu:

```
Telefon tel1 = new Telefon();  
Telefon tel2 = new Telefon();  
tel1.marka="Motorola";  
tel1.numer=123456;  
tel1.kolor="Niebieski";  
tel2.marka="Nokia";  
tel2.numer=789000;
```

```
System.out.println("Marka telefonu 1: "+tel1.marka);  
System.out.println("Numer telefonu 1: "+tel1.numer);  
System.out.println("Kolor telefonu 1: "+tel1.kolor);  
System.out.println("Marka telefonu 2: "+tel2.marka);  
System.out.println("Numer telefonu 2: "+tel2.numer);  
System.out.println("Kolor telefonu 2: "+tel2.kolor);  
tel1.zadzwon(222333);  
tel2.wyslijSMS(333444, "Cześć! To ja!");  
String sms = tel2.odbierzSMS();  
System.out.println("Odebrałem SMS o treści: "+sms);  
tel1.wyslijSMS(tel2.numer, "Jaki ja?");
```

```
Marka telefonu 1: Motorola  
Numer telefonu 1: 123456  
Kolor telefonu 1: Niebieski  
Marka telefonu 2: Nokia  
Numer telefonu 2: 789000  
Kolor telefonu 2: Czarny  
Dzwonię z nr: 123456  
Na nr: 222333  
Wysyłam SMS na nr: 333444  
O treści: Cześć! To ja!  
Odebrałem SMS o treści: Witaj telefonie!  
Wysyłam SMS na nr: 789000  
O treści: Jaki ja?
```

Niektóre obserwacje:

- * Na podstawie klasy można stworzyć wiele obiektów
- * Możemy ustawiać wartości pól, i wywoływać metody
- * Utworzone obiekty są niezależne od siebie - posiadają własne wartości pól, ich metody działają niezależnie

Wartości domyślne

- * Często chcemy ustawić w nowo utworzonym obiekcie jakieś wartości domyślne. Można bezpośrednio po deklaracji przypisać zmiennym domyślne wartości (`String marka="Nieznana";`).
- * Można też utworzyć w klasie odpowiednią metodę, np:

```
public void ustawDaneDomyślne() {  
    marka="Nieznana";  
    model="niewiadomy";  
    numer=0;  
    kolor="czarny";  
}
```

- * Po utworzeniu obiektu, można ją wywołać, wtedy obiekt przyjmie wartości domyślne:
`tel1.ustawDaneDomyślne();`

- * Jeśli teraz odwołamy się do pól, otrzymamy:

```
System.out.println("Marka telefonu 1: "+tel1.marka);  
System.out.println("Numer telefonu 1: "+tel1.numer);  
System.out.println("Kolor telefonu 1: "+tel1.kolor);
```

```
Marka telefonu 1: Nieznana  
Numer telefonu 1: 0  
Kolor telefonu 1: czarny
```

- * Dane domyślne można później zmienić

Konstruktory

- * Zwykle jednak w celu ustawienia wartości domyślnych oraz przeprowadzenia innych operacji wstępnych stosuje się specjalne metody zwane **konstruktorami**.
- * **Konstruktory** mają taką samą nazwę jak klasa, nie umieszczamy w nagłówku słowa **void** ani nie deklarujemy rodzaju zwracanej wartości.
- * Konstruktory są uruchamiane, gdy obiekt jest tworzony, nie trzeba więc wywoływać specjalnie metody.
- * Konstruktor będzie więc wyglądał np. tak:

```
public Telefon() {  
    marka="Nieznana";  
    numer=0;  
    kolor="czarny";  
}
```

Konstruktory, przeciążanie

- * Klasa może posiadać kilka konstruktorów, mają taką samą nazwę, różnią się przekazywanymi parametrami - w ten sposób program „domyśla się” którego konstruktora użyć. Jest to tzw. **przeciążanie metod**. Można więc dopisać konstruktor:

```
public Telefon(String mar, int num, String kol ) {  
    marka=mar;  
    numer=num;  
    kolor=kol;  
}
```

- * Wywołujemy go przy tworzeniu obiektu, przekazując odpowiednie dane:

```
Telefon tel2 = new Telefon("Nokia", 789000, "czarny");
```

Przeciążanie

- * Przeciążanie metod nie ogranicza się wyłącznie do konstruktorów, można je stosować także do innych metod, wywoływana jest ta, do której pasują przekazywane wartości parametrów.

```
public void uruchomAlarm() {  
    System.out.println("Bip, Bip, Bip");  
}  
public void uruchomAlarm(String alarm) {  
    System.out.println(alarm);  
}
```

- * Możemy więc wywołać:

```
tel1.uruchomAlarm();  
tel1.uruchomAlarm("Dryń, Dryń, Dryń");
```

- * Otrzymując kolejno:

```
Bip, Bip, Bip  
Dryń, Dryń, Dryń
```

Zadanie

- * Stwórz program w którym będzie utworzonych kilka obiektów typu **gatunek**
- * Klasa **gatunek** powinna zawierać:
 - * pola przechowujące:
 - * nazwę rodzaju
 - * gatunek
 - * liczbę chromosomów $2n$
 - * podstawową liczbę chromosomów x
 - * uwagi
 - * metody:
 - * podającą pełną nazwę (Rodzaj + gatunek)
 - * podającą haploidalną liczbę chromosomów n
 - * wypisującą wszystkie dane
 - * konstruktory:
 - * tworzący domyślne wartości
 - * pobierający wartości wszystkich pól i ustawiający je
 - * pobierający tylko wartości nazwy rodzaju i gatunku, reszta domyślna
- * W programie powinny być użyte wszystkie metody.