

Programowanie Java

lekcja 1

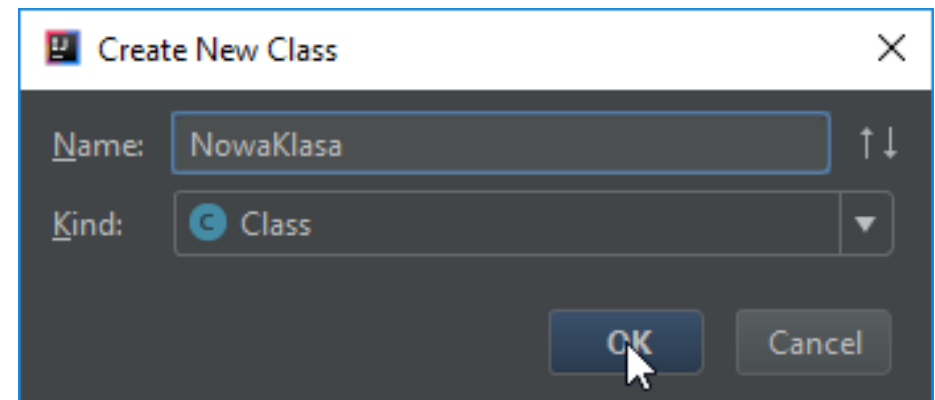
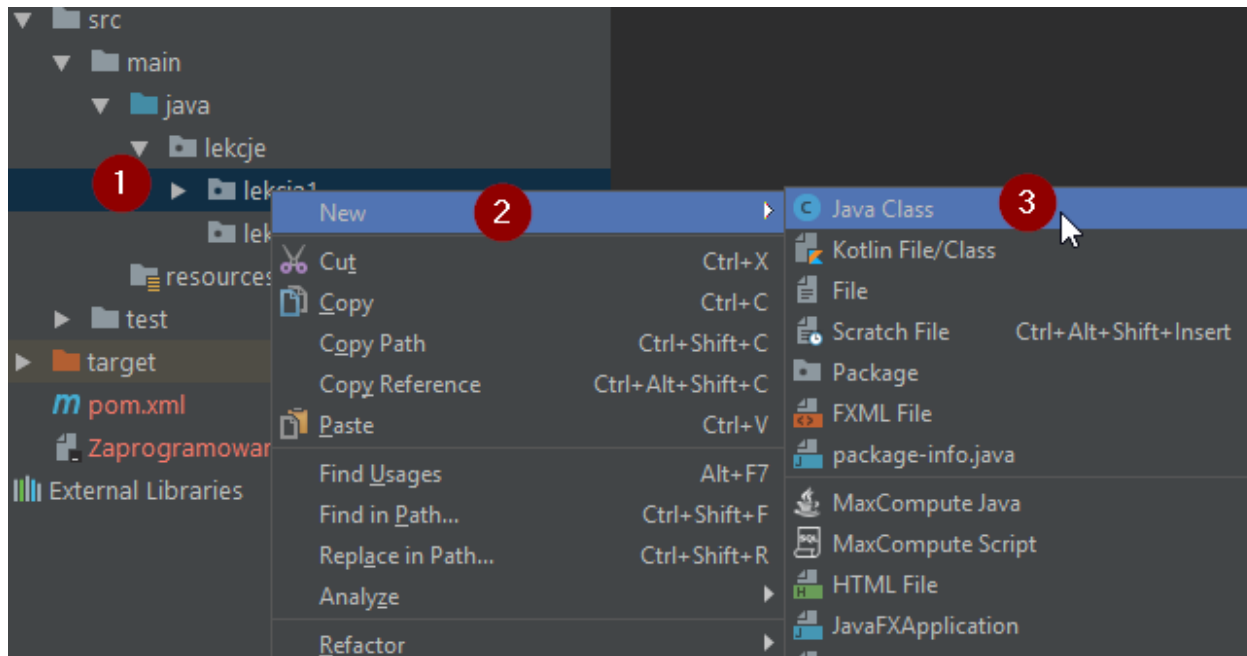
Prowadzący

Sławomir Kaczmarek

Zagadnienia

- **Tworzenie klas**
- **Tworzenie metod**
- **Konstruktor**
- **Enum**
- **Lista**

Przykład 1 – tworzenie klasy



Przykład 1 – tworzenie pól i konstruktora

```
public class Osoba {  
  
    private String imie;  
    private int wiek;  
  
    public Osoba (String _imie, int _wiek){  
        imie = _imie;  
        wiek = _wiek;  
    }  
  
}
```

Pola

Konstruktor

Dzięki konstruktorowi możemy przekazać wartości do pól w trakcie tworzenia obiektu

```
public class Przyklad1App {  
    public static void main(String[] args) {  
  
        Osoba janusz = new Osoba( _imie: "Janusz", _wiek: 17);  
  
    }  
}
```

Przykład 1 – tworzenie metod

```
public class Osoba {  
  
    private String imie;  
    private int wiek;  
  
    public Osoba (String _imie, int _wiek){  
        imie = _imie;  
        wiek = _wiek;  
    }  
  
    public void przedstawSie() {  
        System.out.println("mam na imię: " + imie);  
    }  
}
```

Metoda wypisująca imię



Sposób wywołania metody



```
Osoba janusz = new Osoba( _imie: "Janusz", _wiek: 17);  
janusz.przedstawSie();
```

Zadanie 1 – utwórz klasę Zwierze

Utwórz klasę “Zwierze” posiadającą pola:

- **gatunek**
- **imię**
- **dzwiek (który to zwierze wydaje)**
- **ucieczka (wartość “true” oznacza, że zwierze jest płochliwe)**

Oraz metody:

- **dajGlos()** - wypisanie dźwięku, który wydaje zwierze
- **CzyUcieklPrzedZwierzeciem(Zwierze zwierze)** – przekaż obiekt innego zwierzęcia, sprawdź, czy zwierze jest płochliwe (wartość “true” pola “ucieczka”) I jeśli zwierze jest płochliwe to wypisz gatunek i imię zwierzęcia przed którym uciekł.

Zadanie 1 – sprawdź działanie klasy

Utwórz metodę “main” i sprawdź działanie klasy:

- **Utwórz dwa różne zwierzęta, jedno płochliwe a drugie nie**
- **Przetestuj metodę `dajGlos()`**
- **Przetestuj metodę `czyUcieklPrzedZwierzeciem()`, przekazując do metody obiekt drugiego zwierzęcia**

Przykład 2 – czym jest Enum

Enum jest specjalną klasą, która ogranicza wybór wartości

```
public enum RolaWszkole {  
  
    uczen,  
    nauczyciel,  
    dyrektor;  
  
}
```

Wypisane możliwości
wyboru

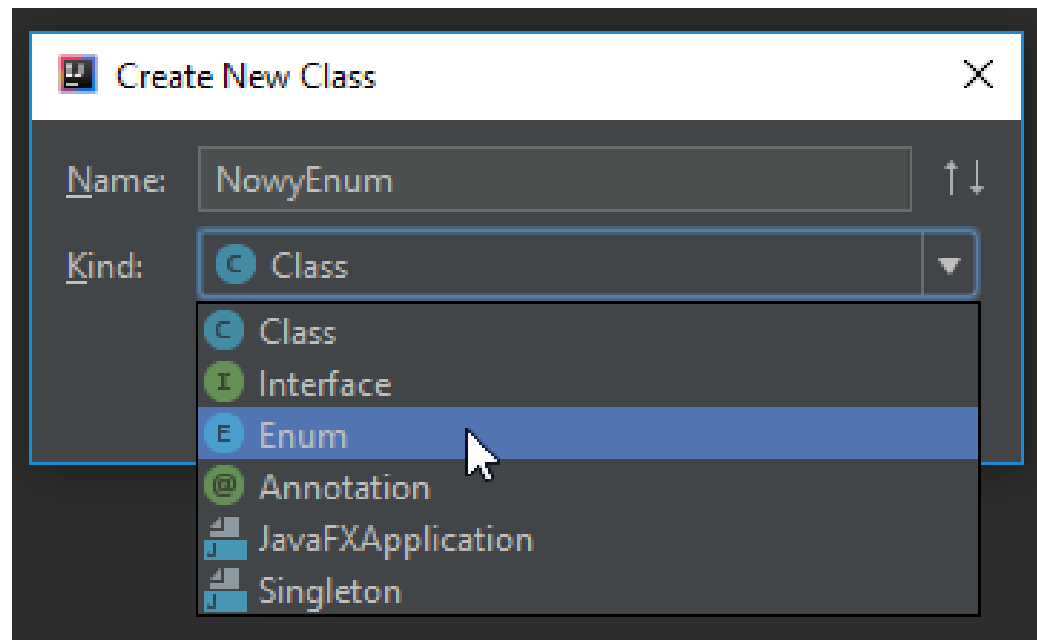
Pole do którego można
przypisać tylko wartości
enuma

Konstruktor przyjmujący
wartość z enuma

```
public class OsobaWszkole {  
  
    private String imie;  
    private int wiek;  
    private RolaWszkole rolaWszkole;  
  
    public OsobaWszkole (String _imie, int _wiek, RolaWszkole _rolaWszkole) {  
        imie = _imie;  
        wiek = _wiek;  
        rolaWszkole = _rolaWszkole;  
    }  
}
```


Przykład 2 – jak utworzyć Enum

Enum stworzymy tak samo jak normalną klasę, jedynie w ostatnim kroku wybieramy w okienku wartość “enum”



Zadanie 2 – dodaj do klasy Zwierzę Enum

Utwórz enum “Dieta” posiadający wartości:

- **miesozerne**
- **roslinozerne**

Do klasy Zwierzę dodaj pole i metodę:

- **pole “dieta”, które przyjmie wartość enuma**
- **metodę zjedz(Zwierze zwierze), zweryfikuj w metodzie, czy zwierze jest mięsożerne i może zjeść inne zwierze, czy nie. Wypisz komunikat: “jestem mięsożerny, chętnie zjem (gatunek)” lub “nie dziękuje, jestem wegetarianinem, nie zjem (gatunek)”**
- **pamiętaj o dodaniu nowego pola do konstruktora!**

Przykład 3 – czym jest Lista

Lista jest zbiorem, który przechowuje obiekty

Podajemy typ obiektów, które będą przechowywane w liście

```
/** Tworzenie obiektów, które dodamy do listy */
Osoba osoba1 = new Osoba( _imie: "Janusz", _wiek: 15);
Osoba osoba2 = new Osoba( _imie: "Krysia", _wiek: 13);

/** Tworzenie listy osób */
List<Osoba> listaOsob = new ArrayList<Osoba>();

/** Dodanie osób do listy */
listaOsob.add(osoba1);
listaOsob.add(osoba2);

/** Metoda size() podaje ilość elementów w liście */
System.out.println(listaOsob.size());

/** Pobranie jednego elementu z listy (numeracja zaczyna się od 0) */
Osoba osobaPobranaZlisty = listaOsob.get(0);
osobaPobranaZlisty.prezentuj();
```

Przykład 3 – zastosowanie Listy

Dzięki liście możemy wyszukać osobę o konkretnym imieniu

```
/** Sprawdzenie, czy lista zawiera obiekt osoba1 */  
  
Boolean czyListaZawieraObiekt = listaOsob.contains(osoba1);  
System.out.println(czyListaZawieraObiekt);  
  
/** Sprawdzenie na którym miejscu w liście znajduje się dany obiekt */  
  
System.out.println(listaOsob.indexOf(osoba1));  
  
/** Wyszukanie w liście osoby o imieniu Janusz */  
  
for(Osoba poszukiwanaOsoba : listaOsob){  
    if(poszukiwanaOsoba.getImie() == "Janusz"){  
        System.out.println("Lista zawiera osobę o imieniu Janusz");  
        System.out.println("Janusz znajduje się na miejscu i indeksie: " + listaOsob.indexOf(poszukiwanaOsoba));  
    }  
}
```

Pętla "for each" pobiera po kolei każdy obiekt z listy. Możemy sprawdzić każdy z tych obiektów, czy posiada imię "Janusz"

Zadanie 3 – utwórz listę Zwierząt

Zmień klasę Zwierze tak, aby gatunek był wybierany z enuma:

- **Utwórz nowy enum “Gatunek” i wpisz kilka wartości**
- **Dostosuj klasę Zwierze**

Utwórz listę i przypisz do niej kilka Zwierząt

- **Pamiętaj, że typ listy musi być zgodny z typem klasy, której obiekty będą przechowywane**
- **Sprawdź długość listy**
- **Wybierz jeden obiekt z listy i wykonaj na nim metodę zjedz(), przekazując do niej inny obiekt z listy**
- **Sprawdź, czy w liście istnieje zwierze o określonym imieniu i wypisz jakiego gatunku jest to zwierze**

Przykład 4 – wartość Enuma jako klucz

Do wartości w enumie możemy przypisać inne wartości

Do produktów przypisano wartości.
Zwróć uwagę na typ liczbowy "double" (kropka zamiast przecinka i "d" dopisane na końcu)

Wartości podane w nawiasie zostaną przypisane
Do pola "cena" przez konstruktor

Konstruktor

Metoda get jak w zwykłej klasie, dzięki której ,
Będziemy mogli zwrócić wartość pola cena

```
public enum Produkty {  
    jablko(1.2d),  
    sok(2.0d),  
    wafelek(2.5d);  
  
    private double cena;  
  
    private Produkty(double _cena) {  
        cena = _cena;  
    }  
  
    public double getCena() {  
        return cena;  
    }  
}
```

Przykład 4 – symulacja Sklepiiku

```
public class Sklepik {  
  
    // Pole przechowujące wartość z enuma  
    Produkty produkt;  
  
    // Konstruktor przypisujący wartość z enuma do pola  
    public Sklepik(Produkty _produkt){  
        produkt = _produkt;  
    }  
  
    // Metoda, do której przekazujemy naszą zapłatę, a ona oblicza,  
    // czy jest wystarczająca i wydaje produkt oraz resztę  
    public void kupProdukt(double zaplata){  
  
        if(zaplata < produkt.getCena()){  
  
            System.out.println("Przykro mi, zaplaciles za malo");  
  
        }else if(zaplata == produkt.getCena()){  
  
            System.out.println("Bardzo prosze, kupiles: " + produkt);  
  
        }else{  
            double resztaDoWydania = zaplata - produkt.getCena();  
            System.out.println("Bardzo prosze, oto Twój produkt: " + produkt + " i " + resztaDoWydania + " reszty");  
        }  
    }  
}
```

Zadanie 4 – symulacja McDonalds'a

Utwórz Enum reprezentujący menu McDonalds

- Wypisz kilka zestawów i dopisz ich ceny w typie “double”
- Pamiętaj o polu przechowującym cenę, konstruktorze i metodzie get

Utwórz klasę McDonalds

- Klasa ma zawierać pola do przechowania klienta (obiekt typu Osoba) oraz produktu.
- Zrób odpowiedni konstruktor
- Zrób metode kup(), która przyjmie jako argument zapłatę, wypisze imię klienta, produkt, który kupił i wyda resztę. Jeśli zapłata będzie za mała to nie wyda produktu

Zadanie 5 – Lista samochodów

Utwórz klasę “Samochod” która zawiera pola:

- numerRejestracyjny – pole typu String
- wlasciciel – pole przechowujące obiekt “wlasciciel”
- marka – pole typu enum

Utwórz pozostałe klasy potrzebne do utworzenia obiektu samochod

Utwórz listę samochodów i wyszukaj w niej samochód:

- po numerze rejestracyjnym
- po właścicielu

Zadanie 5 – Lista samochodów cd.

Znajdź wszystkie samochody BMW:

- **Upewnij się, że masz przynajmniej dwa samochody BMW z różnymi właścicielami**
- **Przypisz je do nowej listy**
- **Wypisz imię właściciela**

Zadanie 6 – Właściciele Zwierząt

Zmodyfikuj klasę Zwierze:

- Dodaj pole przechowujące właściciela zwierzęcia
- Wykorzystaj istniejącą klasę właściciela

Wyszukaj wszystkie zwierzęta jednego z właścicieli:

- Wypisz gatunek oraz imię zwierzaka

Wyszukaj wszystkie zwierzęta jednego gatunku:

- Wypisz imię zwierzaka oraz imię jego właściciela