

Zadania domowe przewidziane na okres 23.07 - 29.07

Kategoria: Klasy, dziedziczenie

Wszystkie zadania mogą zostać zrobione w jednym projekcie. Proszę tylko w klasie Main wykorzystać komentarze, które będą mówiły o tym które zadanie jest teraz wykonywane. Po uruchomieniu programu powinny zostać wykonane wszystkie zadania.

Przykład:

```
public class Main {  
  
    public static void main(String[] args) {  
        //Klasy i dziedziczenie  
        //Zadanie 1  
        ....  
        //Zadanie 2  
        ....  
        //Zadanie 3  
        ....  
        //Listy  
    }  
}
```

WAŻNE!

Wszystkie klasy mają być tworzone w nowych plikach .java

Zadanie 1)

- Stworzyć klasę Zwierzę, która będzie zawierała pola: (String)imię, (Integer)wiek.
- Stworzyć klasę Pies i Kot, które będą dziedziczyły po klasie Zwierze.
- Klasy Pies i Kot nie posiadają swoich własnych pól.
- Klasa Zwierzę powinna posiadać konstruktor publiczny, który przyjmie wszystkie pola opisane powyżej.
- W klasie Zwierze dodać metodę toString()
- W metodzie main stworzyć jeden obiekt klasy Pies i jeden obiekt klasy Kot i wypisać je na ekran.

Zadanie 2)

Rozbudowa Zadania 1.

- Dodać do klasy Zwierzę metodę, która będzie wypisywać na ekran "Jestem Zwierzę".
- Dodać w klasach Pies i Kot nadpisywanie metody(@Override), która została stworzona w klasie Zwierzę. Metoda dla Psa powinna wypisać na ekran "Jestem Pies" a dla Kota "Jestem Kot"
- W metodzie main wywołać dla Psa i Kota tą metodę, którą stworzyliśmy.

Zadanie 3)

Rozbudowa zadań: 1 i 2.

- Stworzyć dwa Enum'y :
 - Pierwszy enum: będzie zawierał rasy psów (dodać po 3 rasy)
 - Drugi enum : będzie zawierał rasy kotów(dodać po 3 rasy)
- Do klasy Pies dodać pole rasa, które będzie zawierało Enum przygotowany dla psa.
- Do klasy Kot dodać pole rasa, które będzie zawierało Enum przygotowany dla kota

Zadanie 4)

Rozbudowa zadań: 1 , 2 i 3

- Stworzyć klasę Człowiek, która będzie zawierała pola, imię, nazwisko
- Klasa człowiek powinna zawierać konstruktor publiczny, który będzie przyjmował wszystkie pola jako argumenty
- Stworzyć klasę Dane Adresowe, która będzie zawierała pola dotyczące adresu, numeru telefonu, maila itp.
- Do klasy Człowiek dodać dwa kolejne pola:
 - Pole zwierzę, które będzie typu Zwierzę
 - Pole dane adresowe, które będzie typu Dane Adresowe
- W metodzie main, stworzyć dwa obiekty klasy Człowiek:
 - Dla jednego obiektu w polu zwierzę przypisać obiekt klasy Pies(który został stworzony we wcześniejszych zadaniach)
 - Dla drugiego obiektu w polu zwierzę przypisać obiekt klasy Kot(który został stworzony we wcześniejszych zadaniach)

Zadanie 5)

Rozbudowa zadań: 1 , 2 ,3 i 4

- Do klasy Człowiek dodać nową metodę która będzie wypisywać : “ Cześć nazywam się [tutaj powinniśmy wypisać imię i nazwisko]”
- Przeciążyć powyższą metodę, która będzie dodawała dodatkowy tekst, który będzie przyjmowany jako parametr metody. Przykład w parametrze metody przeciążonej podamy “Kocham zwierzęta”. Tekst, który zostanie wyświetlony to : “ Cześć nazywam się [tutaj powinniśmy wypisać imię i nazwisko]. Kocham zwierzęta “
- W metodzie main wywołać:
 - metodę w wersji “podstawowej” bez argumentów
 - metodę w wersji “rozszerzonej” z argumentem np. “Kocham zwierzęta”

Kategoria: Listy

Zadanie 6)

Wszystkie poniższe rzeczy wykonujemy w metodzie main

- Stworzyć listę, która będzie zawierała elementy typu String
- Do listy dodajmy 10 elementów wykorzystując metodę add()
- Wyświetlamy listę za pomocą zwykłej pętli for
- Wyświetlamy listę za pomocą pętli foreach

Zadanie 7)

- Stworzyć listę, która będzie zawierała elementy typu Integer
- Do listy dodajemy 15 elementów poprzez wykorzystanie Arrays.asList ()
- Wyświetlamy wszystkie elementy listy za pomocą foreach
- Wyświetlamy wielkość listy
- Usuwamy pierwszy i ostatni element listy
- Wyświetlamy wielkość listy

Zadanie 8)

- Tworzymy klasę Książka, która będzie zawierała pola tytuł, autor, liczba stron, wydawnictwo.
- Tworzymy 11 obiektów klasy Książka
- Dodajemy je do listy, która będzie przechowywać książki(sposób dodania dowolny)
- Wyświetlamy wszystkie elementy za pomocą System.out.println(...)
- Usuwamy środkowy element listy
- Wyświetlamy wszystkie elementy za pomocą System.out.println(...)
- Wyświetlamy wielkość listy
- Przechodzimy po liście(for, foreach) i wyświetlamy tylko książki, które mają powyżej 200 stron

Zadanie 9)

Rozbudowa zadania : 8

- Tworzymy drugą listę która będzie również zawierać książki
- Dodajemy do tej listy 4 obiekty klasy Książka
- Powinniśmy mieć 2 listy jedna z zadania 8, która posiada 10 obiektów i drugą która posiada 4 elementy
- Do listy, która posiada 10 elementów dodajemy naszą drugą listę(która ma 4 elementy). Wykorzystujemy do tego metodę addAll(...)
- Wyświetlamy wielkość listy
- Wyświetlamy wszystkie elementy listy.

Zadanie 10) - *Zadanie nieobowiązkowe - zakres tego programu będzie omawiany dopiero na kolejnych zajęciach.* (Wskazówka wykorzystać zbiory)

Należy napisać program, który będzie pobierał Imiona od użytkownika dopóki użytkownik nie wpisze słowa "END". Program w czasie rzeczywistym powinien weryfikować czy dane Imię już nie zostało niewpisane. Po wpisaniu słowa "END":

- Wyświetl ilość wpisanych imion
- Wyświetl wszystkie elementy.

Zadanie 11) - *Zadanie nieobowiązkowe - zakres tego programu będzie omawiany dopiero na kolejnych zajęciach.*

Napisać program, który będzie przyjmował słowa, które są palindromem(wyszukać w internecie co to jest). Jeżeli wpisane słowo nie jest palindromem, wyświetlić odpowiedni komunikat na ekranie. Słowa w kontenerze nie mogą się powtarzać. Zależy mi w tym zadaniu abyście znaleźli różne metody, które pozwolą sprawdzić czy dane słowo jest palindromem(im więcej tym lepiej). Wpisywanie słów ma się zakończyć jeżeli wpisujemy słowo "END". Po zakończeniu wpisywania wyświetlamy na ekranie wszystkie palindromy, które użytkownik wprowadził.

Dla wszystkich:

- Zanim zaczniecie robić zadania, chciałbym abyście powtórzyli swoją wiedzę z zakresu klas, dziedziczenia i List.
 - [Dziedziczenie](#)
 - [Klasy](#)
 - [Argumenty w metodzie](#)
 - [Przeciążanie metod](#)
 - [Listy](#)
 - [Książka, którą polecam wszystkim](#)

W każdym tym linku(oprócz książki), są małe zadania, które możecie wykonać i sprawdzić jakie jest ich rozwiązanie. Jeżeli chodzi o książkę to mogę wam na następne zajęcia przynieść ją tylko ja posiadam wydanie II.

- Jeżeli znajdą się osoby, które ukończą wszystkie zadania i będą czuły niedosyt. To proszę się o zapoznanie z tymi zagadnieniami:
 - Typy Generyczne
 - Zapis i odczyt plików
 - Klasy abstrakcyjne i polimorfizm
 - Zgłaszanie wyjątków(throw)

- Wszystkie zadania z tej listy mają znaleźć się na publicznym repozytorium [Github](#). Chciałbym od każdego z was dostać link maksymalnie do 30.07.2018.