Cześć. Usiadłam do karnego przerobienia od początku zadania z opiekunami i zwierzętami i chciałabym zrozumieć tą część kodu

private static int nextId=1;

Ok widzę w tym przypadku, że akurat public i private nie wpływa na wynik ale o ile dobrze zrozumiałam wynika to z tego, że mimo, że wynik ostatecznie jest wyswietlany z poziomu innej klasy (Main) to samo w sobie nextId działa tylko w obrębie klasy Animals i bezpośrednio korzysta z niego tylko getter od Id.  (potwierdź proszę czy dobrze myślę)Piszę bo niezbyt rozumiem w jakiej sytuacji powinnam skorzystać z public int nextId a w jakiej z public static nextId ?

public static int nextId=1;   
public int nextId=1;

Widzę skutek: tak sobie puściłam pętle po tablicy typu Animal  Animal[] tablica = new Animal[]{zwierze1, zwierze2, zwierze3, zwierze4}  
to przy public static nextID otrzymałam wynik  
1  Bestia 1 Ania Wittmann 35  
2  Kamikadze 5 Ania Wittmann 35  
3  Bikini 2 Ania Wittmann 35  
4  Wypłosz 5 Ania Wittmann 35z kolei przy public int nextId  
1  Bestia 1 Ania Wittmann 35  
1  Kamikadze 5 Ania Wittmann 35  
1  Bikini 2 Ania Wittmann 35  
1  Wypłosz 5 Ania Wittmann 35staram się zrozumieć co dokładnie stąło się tutaj w obu przypadkach - czy możesz mi tutaj podpowiedzieć co się dzieje? Widzę, że w przypadku public int albo nie dochodzi do podbicia wartosci (nextID++) albo jest ona przywracana do tej pierwotnie ustawionej (w tym przypadku 1)

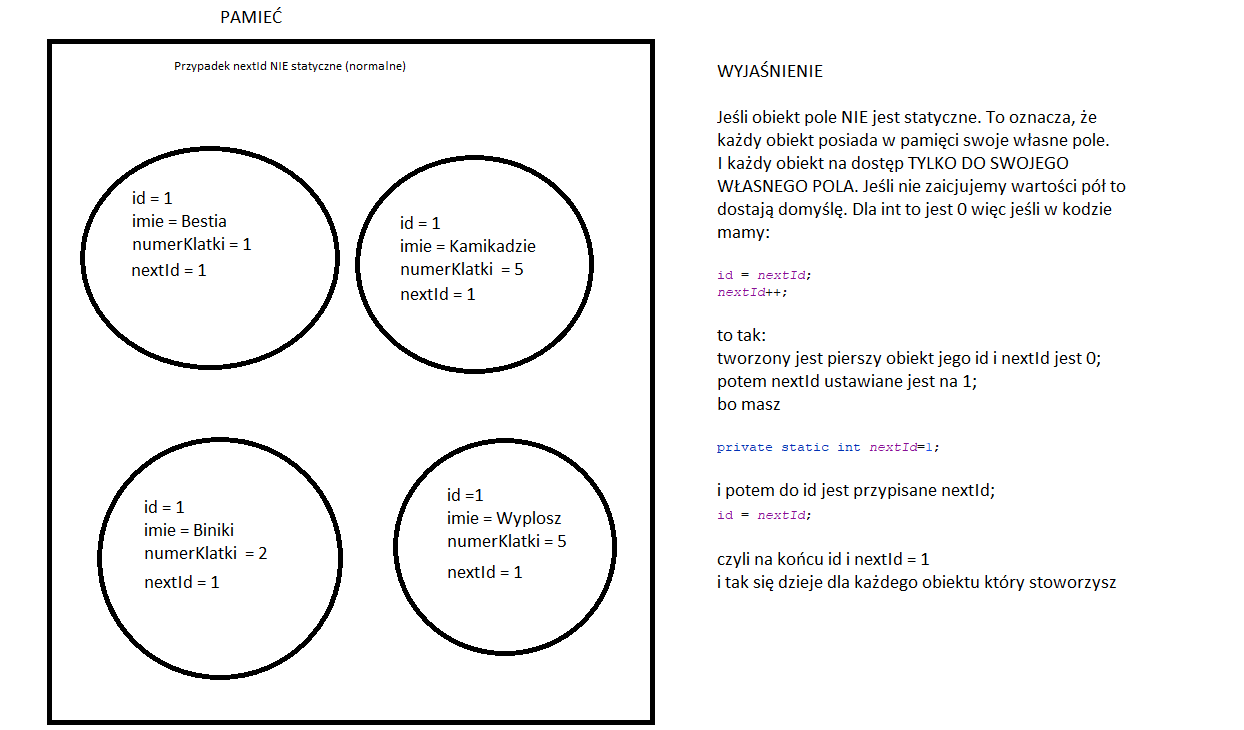
public Animal(String imie, int nrKlatki) {  
 id = nextId;  
 nextId++;  
 this.imie = imie;  
 this.nrKlatki = nrKlatki;  
}public int getId() {  
 id = id;  
 return id;

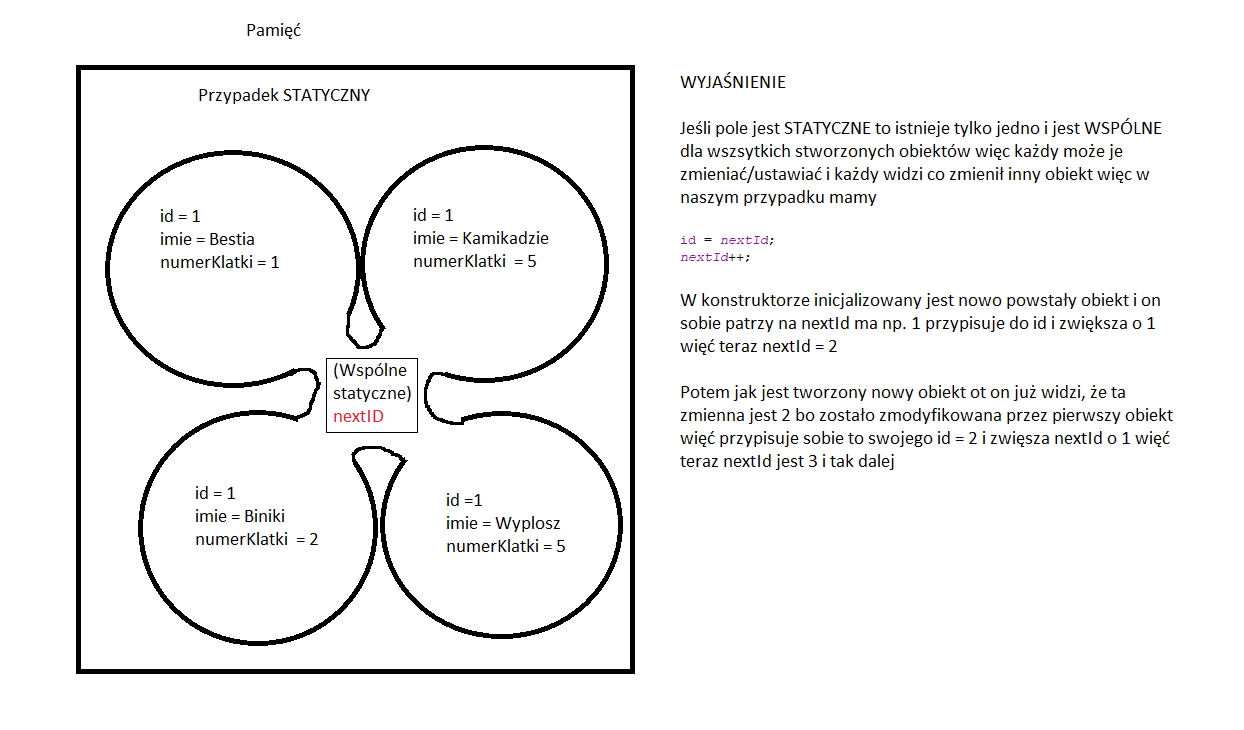
Szukałam odpowiedzi w prezentacji nr 6 i znalazlam na slajdzie 22 ale nie za bardzo umiem to dopasować do tego konkretnego przypadku, więc byłabym wdzięczna za jakieś naprowadzenie ew. może jakiś łopatologiczny, sensownie napisany materiał do analizy własnej (a przynajmniej próby).

[1:25](https://javagda47.slack.com/archives/D01CJH96EB0/p1603581902012600)

I jeszcze drugie pytanie : this w nawiasie - jaka jest jego funkcja ? Pamiętam, że o tym wspomniałeś i  jest możliwe, że nawet to zanotowałam ale chyba  musiałąm to nadpisać bo nie mogę tego znalesc

public void registerAnimal(Animal zwierze) {  
 zwierzeta.add(zwierze);  
 zwierze.setOpiekun(this); // --->   
}





co do tego

[12:29](https://javagda47.slack.com/archives/D01CJH96EB0/p1603625353003700)

public void registerAnimal(Animal zwierze) {  
 zwierzeta.add(zwierze);  
 zwierze.setOpiekun(this); // --->   
}

[12:29](https://javagda47.slack.com/archives/D01CJH96EB0/p1603625361003900)

metoda setOpiekun

[12:29](https://javagda47.slack.com/archives/D01CJH96EB0/p1603625368004200)

powinna zwierzęciu ustawić opiekuna

[12:29](https://javagda47.slack.com/archives/D01CJH96EB0/p1603625389004600)

więc moglibyśmy tak:

[12:30](https://javagda47.slack.com/archives/D01CJH96EB0/p1603625421005200)

Caretaker car = new Caretaker();  
zwierze.setOpiekun(car);

[12:31](https://javagda47.slack.com/archives/D01CJH96EB0/p1603625499006400)

ale w tym przypadku jesteśmy w klasie Caretaker więc this odnosi się do tego Caretaker w którym jesteśmy

[12:32](https://javagda47.slack.com/archives/D01CJH96EB0/p1603625530007000)

i jego chcemy ustawić jako Opiekuna tego zwierzęcia



[**Anna Wittmann**](https://app.slack.com/team/U01ASJ9PACW)  [12:58 PM](https://javagda47.slack.com/archives/D01CJH96EB0/p1603627083013400)

Czyli jeżeli  podsumowujac static oznacza że jest jedno pole dla wszystkich obiektów i one z niego korzystają w relacji 1-n (n obiektów) A jeżeli jest bez statica to mamy 1:1 i każdy obiekt ma swój własny, nie dzielony z innym nextid i wyświetlenie każdego kolejnego obiektu na taki sam wynik tj 0 + 1 =1 i dalsze zwiększenie już nic nie daje bo skonczylismy już z tym obiektem A następny sobie zacznie od początku ze "swoim własnym, tylko swoim" id ?



[**Tomasz Konieczek - trener**](https://app.slack.com/team/U01DEULCKDE)  [12:58 PM](https://javagda47.slack.com/archives/D01CJH96EB0/p1603627136013600)

dokładnie tak