

1 Treść projektu

Utwórz klasę bazową **Ssak** wraz z polami:

- `private String imie;`
- `private short rokUrodzenia;`
- `private boolean mlody;`

oraz, klasy pochodne **Wadera** i **Wilk** dziedziczące po klasie **Ssak**.
Klasa **Wadera** będzie deklarowała pola:

- `private int iloscSzczeniakow;`
- `private Ssak[] szczeniaki;`

natomiast klasa **Wilk** deklaruje pola:

- `private String nazwaStada;`
- `private int pozycja;`

Ponadto każda z klas będzie implementowała metody:

- `String przedstawSie()` - wypisująca wszystkie informacje o danym ssaku jako jedną linię tekstu (wartości pól zostaną oddzielone przecinkami, a linia zostanie zakończona średnikiem);
- `void zapisz(FileOutputStream)` - zapisująca informacje przechowywane w obiekcie;

Uwaga: nie definiuj metod `get` i `set` wykorzystaj właściwości polimorfizmu.

Przygotuj przykład z dwoma oddzielnymi stadami wilków, składającymi się z takiej samej ilości samców i samic, wśród których jest przynajmniej $\frac{1}{3}$ młodych.

Następnie zapisz informacje o osobnikach w stadzie do pliku: "wilki.txt". Uruchamiając program należy dokonać sprawdzenia czy plik o nazwie "wilki.txt" istnieje, jeżeli tak należy użyć metody `public static Ssak[] zaladuj(String sciezka)` pozwalającą odczytać i odtworzyć stado. Natomiast gdy nie istnieje, program utworzy stado i zapisze ich stan osobniczy do pliku.

2 Wymagania formalne

Projekt należy oddać do dnia 28.VI.2020

Prace oddawane w ramach projektu powinny być dostarczone w postaci pojedynczego pliku `.java` o strukturze zgodnej z poniższym schematem:

```
1 public
2     class SXXXXXX_p03 {
3
4         public static void main(String[] args) {
5             // ... tresc programu testowego ...
6         }
7
8         // zestaw metod własnej implementacji studenta
9     }
10
11 class ABC{
12     // zestaw metod własnej implementacji studenta
13 }
```

gdzie jako XXXXXX należy podać numer studenta.

Powstały (jako rozwiązanie) plik, należy dostarczyć jako załącznik do projektu w usłudze *Teams*. A następnie zwrócić prowadzącemu przez naciśnięcie przycisku "Zwróć" (ang. "Turn in") przed upływem terminu wyszczególnionego w treści projektu.

Realizacja projektu wymaga deklaracji obiektów wraz z inicjacją pól, wykorzystania metod, deklaracji i operacji na tablicach, instrukcji warunkowych i pętli. W projekcie nie można używać, kolekcji oraz struktur danych, które nie są autorstwa oddającego lub nie zostały wyszczególnione w treści projektu jako dozwolone.

Projekt nie może zostać poddany żadnym formom zaciemniania kodu.

Projekt nie oddany (zwrócony) do oceny prowadzącego, przed upływem wskazanego w nim terminu skutkuje uzyskaniem **zerowej** liczby punktów.

Niespełnienie któregośkolwiek z wymogów formalnych opisanych powyżej, skutkować będzie **zerową** liczbą punktów.