

GUI III

Typ wyliczeniowy (*enum*) jako klasa, Comparable, Comparator, Generics

1. Stwórz klasę wyliczeniową *KategoriaWagowa* posiadającą cztery wartości, tj.: *lekka*, *srednia*, *polCiezka*, *ciezka*. Klasa ta ma posiadać dwa pola typu zmiennoprzecinkowego: *wagaMin*, *wagaMax*. Ponad to musi posiadać konstruktor, gettery i *toStringa* (wykorzystującego tylko metodę *this.name()*).

Nazwa	Waga od	Waga do
<i>lekka</i>	0.0	61.0
<i>srednia</i>	61.01	79.0
<i>polCiezka</i>	79.01	90.7
<i>ciezka</i>	90.71	Double.MAX_VALUE

2. Stwórz klasę *Bokser* posiadającą pola:

String nazwisko,

double waga, *wzrost*,

KategoriaWagowa kategoriaWagowa

W klasie tej napisz konstruktor inicjalizujący wszystkie pola tej klasy, jednakże kategoria wagowa powinna zostać przypisana poprzez wywołanie prywatnej metody *przypiszKategorieWagowa()*, która na podstawie wagi dostarczonej do konstruktora dobierze (zwróci) odpowiednią kategorię wagową dla boksera. Napisz również *toStringa*.

3. Zainicjalizuj 2-3 obiekty typu *Bokser* (najlepiej tak aby byli w innych kategoriach wagowych). Następnie stwórz nową listę tablicową, przechowującą obiekty klasy *Bokser*. Dodaj do tej listy swoich bokserów oraz dodaj jeszcze jednego bez przypisywania go do obiektu (metoda *add()* i w środku *new Bokser(...)*).
4. Wydrukuj na konsoli za pomocą pętli *forEach* (taki *for* z dwukropkiem) zawartość listy z bokserami. Za pomocą metody *sort()* z klasy *Collections* posortuj bokserów w liście rosnąco pod względem wagi. Ponownie wydrukuj

stan listy na konsoli. Podpowiedź: klasa *bokser* musi implementować interfejs *Comparable* i definiować (implementować) działanie metody *compareTo()*.

5. Bez nanoszenia zmian w klasie *Bokser*, ponownie posortuj tę listę za pomocą dwuargumentowej metody *sort* z klasy *Collections*, pod względem wzrostu bokserów malejąco. Podpowiedź: anonimowy obiekt interfejsu *Comparator* ze zdefiniowaniem metody *compare*.
6. Stwórz klasę *MyClass* posiadającą jedno prywatne pole dowolnego $\langle T \rangle$ typu. Klasa ta ma posiadać konstruktor i metodę *showType()*, która to wywołana na rzecz obiektu tej klasy, ma wydrukować na konsoli informację o nazwie typu wyżej wspomnianego pola w obiekcie.
7. Stwórz klasę *NumericOperations*, która będzie posiadała jedno prywatne pole dowolnego numerycznego typu. Stwórz metodę *suqare()* podnoszącą wartość pola tegoż obiektu do kwadratu (ma zwracać *double*). Zaprezentuj działanie.
8. W tej samej klasie co w zadaniu poprzednim zdefiniuj metodę *isEqual* zwracającą wartość logiczną, a mającą za zadanie porównać do siebie wartość pola obiektu *NumericOperations* na rzecz którego została wykonana z tym samym polem w innym obiekcie *NumericOperations* dostarczonym jako argument do tej metody. Zaprezentuj działanie.