

Zadanie 1

1. Stwórz klasę Member w pakiecie zadania.zadanie1, zawierającą prywatne pola:
 1. String name
 2. String email
 3. Integer id;
2. Dodaj publiczny konstruktor ustawiający te 3 wartości
3. Dodaj trzy metody publiczne typu get (akcesory), zwracające poszczególne pola klasy
4. Stwórz klasę MainClass, w ramach której znajduje się metoda main, sprawdź poprawność działania powyższych metod
5. Stwórz klasę Team, zawierającą pola:
 1. String name
 2. Member[] members
6. Stwórz konstruktor, który będzie mógł być wykorzystany w celu stworzenia drużyny. Niech przyjmuje parametr name, oraz parametr typu varargs np. (Member ... members)
7. Stwórz metody klasy Team, które zwracają ilość osób w drużynie, oraz tablicę członków tej drużyny
8. w klasie MainClass, utwórz kilku (3-4) członków drużyny, a następnie przy ich pomocy, stwórz obiekt typu Team. sprawdź działanie metod z punktu 7. Wypisz zawodników przy pomocy pętli for-each lub for, bazując na tablicy

Zadanie 2

1. Stwórz klasę Point2D, która zawiera dwa typu double: x oraz y;
2. Utwórz dwa konstruktory obiektów tej klasy:
 1. jeden będzie ustawiał pola x oraz y na podstawie parametrów przekazanych przez użytkownika jako parametry konstruktora
 2. drugi będzie ustawiał pola x oraz y na podstawie wartości pól wyciągniętych z obiektu tej samej klasy, podanej jako parametr konstruktora
3. Utwórz metody:
 1. zwracające odpowiednio wartości pól x oraz y (getter / akcesor)
 2. wyświetlające obiekt w formie string (toString)
 3. porównujące obiekty między sobą (equals)
4. Utwórz kilka obiektów tej klasy wykorzystując oba konstruktory
5. Stwórz klasę Point3D, która rozszerza klasę Point2D rozszerzając ją o jeden dodatkowy parametr: double z
6. Dodaj do niej konstruktory bazujące na konstruktorach klasy point2D (konstruktor super())
7. Nadpisz metody toString oraz equals dla klasy Point3D

Zadanie 3

1. Stwórz klasę `Osoba`, która zawiera następujące pola:
 1. Imie
 2. Nazwisko
 3. Pesel (możesz użyć programu do walidacji peselu, aby sprawdzić czy podany pesel jest poprawny ;)
2. Zdefiniuj konstruktor dla tej klasy, przyjmujący 3 wspomniane wyżej parametry oraz metody
 1. `przedstawSie()` zwracający imię i nazwisko w formie `String`
 2. nadpisz metodę `toString()` z klasy `Object`
 3. nadpisz metodę `equals()` z klasy `Object`
3. Utwórz klasę `Uczen`, dziedziczaca po klasie `Osoba`
4. Utwórz klasę `UczenSzkoły`, dziedziczącą po klasie `Uczen`,
 1. klasa zawiera dodatkowe pole `nazwaSzkoły`
 2. posiada konstruktor wykorzystujący konstruktor klasy nadrzędnej
 3. klasa zawiera metodę `zmienSzkołę()`, która pozwala na zmianę szkoły przez ucznia (wartości pola `nazwaSzkoły` w klasie `UczenSzkoły`)
 4. nadpisz metody `toString()` oraz `equals()`
 5. dla ambitnych: stwórz oddzielną klasę `Szkola` reprezentującą szkołę
5. Utwórz klasę `UczenKlasy`, który dziedziczy po klasie `UczenSzkoły`
 1. klasa zawiera dodatkowe pole : `klasa`
 2. wykorzystuje konstruktor klasy nadrzędnej
 3. posiada metody `getKlasa()` pozwalającą pobrać nazwę klasy, oraz `zmienKlasę()` zmieniającą wartość pola `klasa` w `UczenKlasy`
 4. nadpisz metody `toString()` oraz `equals()`
6. Dla klasy `UczenKlasy`, stwórz konstruktory, które pozwalają stworzyć obiekt `UczenKlasy`, wykorzystując parametry konstruktora typów: `UczenSzkoły`, `Uczen` oraz `Osoba` + dodatkowe wymagane parametry

Zadanie 4

1. Utwórz klasę `Drzewo` z polami `boolean wiecznieZielone`, `int wysokosc` oraz `String przekrojDrzewa`, następnie zaimplementuj konstruktory inicjujące pola i metodę `String toString()` wypisującą na konsoli informacje o drzewie.
2. Utwórz klasę `DrzewoIglaste` dziedziczącą po klasie `Drzewo`, oraz definiującą pola `int iloscIgiel` i `double dlugoscSzyszki`. Następnie przygotuj konstruktor inicjujący wszystkie pola oraz metodę `String toString()` wypisującą na konsoli informacje o drzewie wraz z wartościami pól opisujących drzewo iglaste.
3. Utwórz klasę `DrzewoLisciaste` dziedziczącą po klasie `Drzewo`, oraz definiującą pole `int`

ksztaltLiscia i metodę `String toString()` wypisującą na konsoli wszystkie informacje o drzewie.

4. Utwórz klasę `DrzewoOwocowe` dziedziczącą po klasie `DrzewoLisciaste`, oraz definiującą pole `String nazwaOwoca`, podobnie jak w poprzednich zadaniach uzupełnij klasę o konstruktor i funkcjonalną metodę `toString`.
5. Na koniec utwórz program (metoda `main`) w klasie `MainClass`, który wypełni tablicę `Drzewo[]` las drzewami takimi jak: sosny, modrzewie, dęby, osiki, morelowce czy śliwy. Wykorzystując mechanizm polimorfizmu wyświetl wszystkie informacje o wszystkich drzewach (pętla `foreach`, lub zwykły `for`)