

Kolokwium przykładowe 2023

11 stycznia 2023

Uwagi wstępne:

W katalogu domowym użytkownika `student` utwórz katalog o nazwie `NazwiskoImie` (nie używaj polskich liter ani spacji). **Jakakolwiek** próba **ściągnięcia** kończy kolokwium z punktacją 0pt, bez możliwości poprawy.

UWAGA: Program, który **się nie kompiluje** traktuje się jako **niezaliczony**.

Klasa `Mebel`

Zdefiniuj klasę `Mebel`. Zadeklaruj w tej klasie prywatne pole `nazwa` typu `String`, prywatne pole `dlugosc` typu `double`, prywatne pole `szerokosc` typu `double`, oraz prywatne pole `iloscNog` typu `Integer` oraz statyczne prywatne pole `ile` typu `int`. Nie definiuj w tej klasie konstruktora domyślnego. Następnie:

1. napisz czteroargumentowy konstruktor klasy `Mebel`, którego pierwszy argument służy do inicjalizacji pola `nazwa`, a drugi argument służy do inicjalizacji pola `dlugosc`, a trzeci argument służy do inicjalizacji pola `szerokosc`, a czwarty argument służy do inicjalizacji pola `iloscNog`. W konstruktorze tym zwiększaj wartość pola `ile` o 1.
2. napisz trzyargumentowy konstruktor klasy `Mebel` wykorzystujący powyższy konstruktor klasy `Mebel` do zainicjalizowania pola `nazwa` nazwą `Jakis Mebel`.
3. zdefiniuj metody typu `get` dla wszystkich pól.
4. nadpisz metodę `toString` z klasy `Object` zwracającą łańcuch zawierający nazwę klasy, a po niej (w nawiasach kwadratowych i rozdzielone przecinkami) wartości pól klasy. Jeżeli pole `nazwa` ma wartość `Jakis Mebel` nie wypisuj jej.
5. nadpisz metodę `equals` z klasy `Object`. W porównaniu uwzględnij wszystkie pola klasy.
6. zdefiniuj statyczną metodę `getIle` zwracającą wartość pola `ile`.

Klasa `Biurko` dziedzicząca po klasie `Mebel`

W tej klasie zadeklaruj prywatne pole `dataProdukcji` typu `LocalDate` oraz prywatne pole `przekatnaMonitora` typu `double`. Nie definiuj w tej klasie konstruktora domyślnego. Następnie:

1. napisz pięcioargumentowy konstruktor klasy `Biurko`, którego pierwszy argument służy do inicjalizacji pola `nazwa`, a drugi argument służy do inicjalizacji pola `dlugosc`, trzeci argument służy do inicjalizacji pola `szerokosc`, czwarty argument służy do inicjalizacji pola `iloscNog`, a piąty argument służy do inicjalizacji pola `przekatnaMonitora`. Pole `dataProdukcji` zainicjalizuj datą pobraną z systemu operacyjnego.

2. zdefiniuj dla pola `dataProdukcji` metodę typu `set` (argumenty to rok, miesiąc i dzień).
3. zdefiniuj dla pola `dataProdukcji` metodę typu `get`.
4. nadpisz metodę `toString` z klasy `Mebel`. Metoda `toString` powinna zwracać łańcuch będący konkatencją łańcucha zwracanego przez metodę `toString` nadklasy oraz łańcucha zawierającego nazwę klasy, a po niej (w nawiasach kwadratowych) datę gwarancji w formacie `yyyy-MM-dd`, dzień tygodnia w którym został wyprodukowany, dzień roku w którym został wyprodukowany.
5. zdefiniuj metodę `equals` wykorzystując metodę `equals` z nadklasy.

Klasa Main

W funkcji `main` klasy `Main`:

1. utwórz obiekt klasy `Mebel` o nazwie `maly` z wartością pola `nazwa` równą `Maly Mebel` oraz wartością pola `dlugosc` równą 1, wartością pola `szerokosc` równą 0.5, wartością pola `iloscNog` równą 4.
2. wypisz dane obiektu `maly` na ekran w postaci zwracanej przez metodę `toString`.
3. wypisz wynik zwracany przez funkcję `getIle` wywołaną na rzecz klasy `Mebel`.
4. utwórz obiekt klasy `Mebel` o nazwie `sredni` z domyślną wartością pola `nazwa` oraz wartością pola `dlugosc` równą 2, wartością pola `szerokosc` równą 1, wartością pola `iloscNog` równą 4.
5. wypisz dane obiektu `sredni` na ekran w postaci zwracanej przez metodę `toString`.
6. utwórz obiekt klasy `Biurko` o nazwie `maleBiurko` z wartością pola `nazwa` równą `Male Biurko` oraz wartością pola `dlugosc` równą 2, wartością pola `szerokosc` równą 1, wartością pola `iloscNog` równą 4 i wartością pola `przekatnaMonitora` równą 22.
7. w obiekcie `maleBiurko` zmień pole `dataProdukcji` na ostatni dzień lutego 2005 roku.
8. wypisz dane obiektu `maleBiurko` w postaci zwracanej przez metodę `toString`.
9. wypisz wynik zwracany przez funkcję `getIle` wywołaną na rzecz klasy `Mebel`.
10. utwórz dwuwymiarową tablicę `spis`, w której będzie można umieszczać obiekty klas `Mebel` oraz `Biurko`.
11. w pierwszym wierszu tablicy `spis` umieść obiekt `sredni` i obiekt `maly`, w drugim wierszu tablicy umieść obiekt `maleBiurko` i obiekt `sredni`.
12. wypisz tablicę dwuwymiarową `spis`
13. utwórz tablicę `spisNazw`, w której będzie można umieszczać wartości pola `nazwa`.
14. umieść wartości pola `nazwa` w tablicy `spisNazw` w podanej kolejności: `sredni`, `maly`, `maleBiurko` i `sredni`.
15. stosując instrukcję `foreach` wypisz wszystkie obiekty z tablicy `spis`.
16. wypisz wyniki porównań przy pomocy metody `equals` obiektu `sredni` z obiektem `Maly Mebel` oraz z obiektem `maleBiurko`.

17. policz ilość biurek w tablicy `spis` nie korzystając z informacji o tym jakie obiekty umieściłeś uprzednio w tablicy `spis`.
18. wypisz nazwy klas obiektów z tablicy `spis` (oddzielając je przecinkami) nie korzystając z informacji o tym jakie obiekty umieściłeś uprzednio w tablicy `spis`.