

Przykładowy egzamin - Zestaw E16

Ostatnia aktualizacja pliku: 21.01.2024 23:03.

Imię i nazwisko, numer albumu

Informacje wstępne

- Punktacja: 46-50 pkt - bdb(5,0); 41-45 pkt - db+(4,5); 36-40 pkt - db(4,0); 31-35 pkt - dst+(3,5); 26-30 pkt - dst(3,0); 0-25 pkt - ndst (2,0).
- **Egzamin należy wykonać na komputerach zamontowanych na stałe w pracowniach.**
- Student przysyłając rozwiązania oświadcza, że rozwiązał je samodzielnie.
- W trakcie egzaminu nie można korzystać z żadnych materiałów pomocniczych w żadnej formie. Wszelkie kody powinny być napisane manualnie bez wspomagania się dodatkami automatycznie generującymi kod (np. Copilot, chat GPT itp.).
- Publikowanie poleceń i rozwiązań w internecie jest zabronione do czasu napisania egzaminu przez wszystkich.
- Należy zwracać uwagę na właściwe umieszczenie kodu (luzem lub w pakiecie). Kod musi się kompilować, aby był sprawdzany. Kod zakomentowany nie będzie sprawdzany.
- Należy oddzielać klasę z definicjami od klasy testującej (z main) zgodnie z poleceniami.
- Jeśli w poleceniu nie jest podany typ zmiennej, można go wybrać dowolnie.
- Jeśli w danej metodzie nie ma sprecyzowanej „walidacji”, to można ją pominąć.
- **Metody nie powinny wykonywać nadmiarowych, nielogicznych czynności.**
- Poza zmiennymi/polami w klasie wymienionym w poleceniach zabronione jest tworzenie innych pól w klasie. Stworzenie dodatkowych metod jest dopuszczalne (o ile polecenie tego nie zabrania), ale nie należy tego nadużywać.
- Należy zachowywać kolejność argumentów w konstruktorach i metodach. Należy dążyć do tego, że nazwy argumentów metod powinny pokrywać się z nazwami pól w klasie, gdzie to ma sens.
- Warto zwracać uwagę na typ zwracany metod — jeśli metoda ma „coś” zwrócić, będzie to wskazane w poleceniu.
- Jeśli w poleceniu nie są sprecyzowane modyfikatory dostępu, należy dostępować zgodnie z zasadami hermetyzacji.
- Jeśli w poleceniu pojawia się informacja o konieczności zachowania formatowania napisów (np. wielkość znaków, znaki interpunkcyjne), to należy to bezwzględnie wykonać.
- **W rozwiązaniach należy uwzględniać dobre praktyki omawiane na wykładzie, o ile polecenie nie mówi coś innego.**
- Rozwiązania (projekt z IntelliJ) należy w całości spakować jako archiwum zip. Następnie ustawić nazwę. Rozwiązania należy umieścić na pendrive przekazany przez prowadzącego egzamin. Rozwiązania niespakowane jako zip nie będą sprawdzane. Archiwum powinno być bez hasła.
- **Nazwa archiwum powinna być wg schematu NUMERZESTAWU_NUMERALBUMU.zip gdzie numer zestawu znajduje się na górze kartki z poleceniami. np. A23_123456.zip.**
- Zawartość pendrive będzie pusta. Umieszczenie poleceń na pendrive powinno odbyć się w czasie egzaminu. Rozwiązania po czasie mogą nie być sprawdzane.
- Podpunkty będą oceniane kaskadowo oraz wykładniczo — wykonanie ich bez wykonania wcześniejszych podpunktów może oznaczać zero punktów. Koniec polecenia ma największą wagę w ocenie danego zadania.
- O ile nie zaznaczono w poleceniu inaczej, każdą z metod należy wywołać co najmniej jeden raz (może być bardzo trywialnie). Warto zwrócić uwagę, że samo tworzenie obiektów w każdym zdefiniowanym samodzielnie typie nie jest wymagane (chyba że polecenie tego wymaga).
- Po kartkach z poleceniami można pisać i traktować jako brudnopis.

Zadanie 1. (13pkt max.)

A. Wykonaj poniższe czynności:

- Stwórz klasę `Hotel` (pol. `Hotel`), która powinna być umieszczona w pakiecie `tourism`.
- Klasa powinna posiadać prywatne pola:
 - `name`, (nazwa hotelu), typ `String`.
 - `capacity`, (pojemność hotelu, czyli liczba dostępnych pokoi), typ `double`.
- Napisz dwuargumentowy konstruktor tej klasy. Kolejność argumentów powinna być taka sama jak wyżej. Zapewnij niezależnie warunki sprawdzające poprawność:
 - Nazwa nie może być nullem - jeśli jest nullem, to zamień ją na napis pusty (`""`).
 - Pojemność musi być liczbą dodatnią, w przeciwnym wypadku ustaw ją na `50.0`.
- Nadpisz metodę `toString` tak, aby zwracała napis z reprezentacją obiektu.
- Zaimplementuj w tej klasie metodę `clone` z interfejsu `Cloneable`.

B. Wykonaj poniższe czynności:

- Stwórz klasę `TestHotel` (klasa testująca) umieść w pakiecie tym samym co klasy z punktu A, ale w innym pliku. Umieść w tej klasie tylko metodę `main`. W metodzie `main` stwórz 2 obiekty typu `Hotel` i sprawdź działania klonowania (kopiowania) - zmień nazwę oryginału, a następnie sprawdź nazwy obu obiektów.

Zadanie 2. (13pkt max.)

A. Utwórz rekord `BusRoute` w pakiecie `transport`, który powinien zawierać trzy pola:

- `routeNumber`: typu `String`, reprezentującego numer trasy autobusowej.
- `startLocation`: typu `String`, reprezentującego punkt początkowy trasy.
- `endLocation`: typu `String`, reprezentującego punkt końcowy trasy.

B. Dodaj do rekordu `BusRoute`:

- Kompaktowy konstruktor, który weryfikuje, czy trasa (`routeNumber`) nie jest pusta. Jeśli `routeNumber` jest pusty lub `null`, konstruktor powinien rzucać wyjątek `IllegalArgumentException`.
- Metodę `isLongRoute`, która zwraca `true`, jeśli trasa autobusu jest uznawana za długą (długość napisu `routeNumber` jest większa niż 4), i `false` w przeciwnym przypadku.
- Metodę `printDetails`, która wyświetla informacje o trasie, włączając w to numer trasy, punkt początkowy i końcowy.

C. W pakiecie `transport`, utwórz klasę testową `TestBusRoute` z metodą `main`, w której:

- Utwórz dwa obiekty typu `BusRoute` z różnymi danymi.
- Wywołaj metodę `printDetails` na każdym z obiektów, aby wyświetlić ich szczegóły.
- Sprawdź, które trasy są długie, używając metody `isLongRoute`.

Zadanie 3. (12pkt max.)

- Wykonaj czynności w pakiecie `swap`.
- Utwórz statyczną metodę generyczną `swapElements`, która przyjmuje tablicę elementów typu generycznego `T` oraz dwa indeksy w tej tablicy. Metoda powinna zamieniać miejscami elementy znajdujące się na tych indeksach.

- Na przykład, `swapElements(new String[]{"apple", "banana", "cherry"}, 0, 2)` powinno zwrócić tablicę `{"cherry", "banana", "apple"}`.
- Napisz klasę `VideoGame`, która zawiera pola: `name` (typu `String`), `developer` (typu `String`) i `rating` (typu `float` reprezentującego ocenę gry).
- Stwórz przypadek testowy, aby zademonstrować działanie metody na niepustej tablicy obiektów typu `VideoGame`.

Zadanie 4. (12pkt max.)

- Poniższe czynności wykonaj w pakiecie `finding`.
- Zaimplementuj statyczną metodę generyczną `findValueByKey`, która otrzymuje `HashMap<K, V>` oraz klucz typu `K`. Metoda powinna zwracać wartość skojarzoną z podanym kluczem. Jeśli klucz nie istnieje w mapie, metoda powinna zwracać `null`. Pamiętaj, aby obsłużyć potencjalne wyjątki związane z przekazaniem `null` jako argumentu. Stwórz przypadek testowy. .

