

Otwarto: środa, 23 listopada 2022, 11:20

Wymagane do: środa, 23 listopada 2022, 12:50

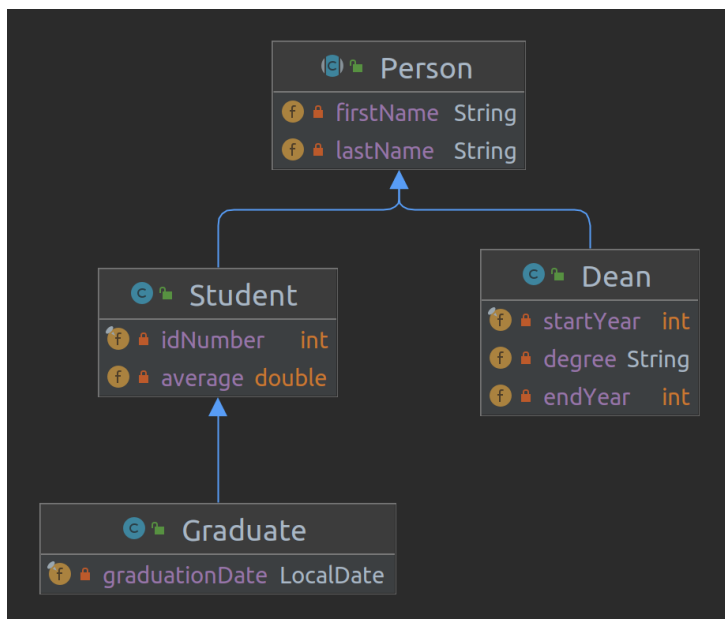
Proszę pobrać i rozpakować archiwum. W archiwum znajdują się testy jednostkowe: po jednej klasie testującej dla każdej z klas, które trzeba napisać w 1. zadaniu.

Wszystkie pola pisanych dziś klas powinny być **prywatne**. Proszę też **korzystać z napisanych przez siebie metod** oraz **dziedziczenia** – jeśli dana funkcjonalność jest już zaimplementowana, to nie powtarzamy jej (np. jeśli klasa *Person* potrafi zwrócić poprzez *toString()* napis zawierający imię i nazwisko, to niech klasy potomne to wykorzystują). Niezastosowanie się do tych wymagań będzie miało wpływ na ocenę (3 pkt.).

1. Zaimplementuj:

- (1 pkt.) abstrakcyjną klasę *Person* z abstrakcyjną metodą `abstract public Person greater(Person ob)`,
- (1.5 pkt.) klasę *Student*, rozszerzającą *Person* (metoda *greater* zwraca obiekt reprezentujący studenta o wyższej średniej),
- (1.5 pkt.) klasę *Graduate*, rozszerzającą *Student*,
- (1.5 pkt.) klasę *Dean*, rozszerzającą *Person* (metoda *greater* zwraca obiekt reprezentujący dziekana, który dłużej sprawował swoją funkcję).

Klasy należy tak zaprojektować, aby były **zgodne z załączonym diagramem UML** oraz **poprawnie przechodziły wszystkie testy** zawarte w archiwum. Wskazówka: muszą należeć do odpowiedniego **pakietu**. Z testów wynika, jakie **konstruktory/metody** muszą być zawarte w klasach.



2. (1.5 pkt.) Poza pakietem w osobnej klasie wykorzystaj klasy z 1. zadania do utworzenia jednej wspólnej tablicy, której elementami będą obiekty klas *Student*, *Graduate* i *Dean*. Przy pomocy jednej pętli for-each wypisz zawartość tej tablicy.

Dzisiejsze klasy proszę przesłać w archiwum z zachowaniem struktury katalogów!

Status przesłanego zadania

Status przesłanego zadania	Przesłane do oceny
Stan oceniania	Ocenione
Pozostały czas	Zadanie zostało złożone 13 min. 19 sek. przed terminem