

Zadanie znajduje się na repozytorium. Należy sklonować je na swój komputer, utworzyć nowy własny branch nazwany <Nazwisko\_Imię> i wszelkie commity robić właśnie na ten branch. Po skończonej pracy wszystko należy wypchnąć na zdalne repozytorium (git push).

Dana jest hierarchia klas reprezentujących pracowników pewnej fabryki. Jest też klasa, która generuje listę tych pracowników. Pracownicy ci są od czasu do czasu wysyłani na różnego typu „wycieczki”, np. do innych fabryk. Tego typu „wycieczki” trzeba najpierw odpowiednio zorganizować i w tym celu należy ocenzurować niektóre informacje o danym pracowniku. Nie można tego zrobić za pomocą normalnego kodu, ponieważ struktura tych wycieczek i kryteria tego co należy ukryć zmieniają się przy każdej z takich wycieczek. Należy więc zrobić to za pomocą odpowiednich adnotacji, które można dołączyć elementów kodu, oraz mechanizmu refleksji.

Należy stworzyć odpowiednie adnotacje i zaadnotować nimi odpowiednie klasy, metody i pola. W projekcie istnieje klasa cenzorująca, implementująca interfejs **Cenzorable**, który ma jedną metodę **cenzuruj**. Metoda ta przyjmuje listę pracowników, skanuje ją pod kątem występowania odpowiednich adnotacji i zwraca tą listę pracowników, ale już z ocenzurowanymi informacjami. **Należy zaimplementować tą metodę w klasie cenzorującej zgodnie z podpunktami poniżej.**

Przykład:

Znaleziono adnotowane pole typu String z adnotacją @SkasujZawartosc. W konsekwencji, w to pole wstawiana jest wartość pusta.

W **Main** tworzona jest lista pracowników, poddawani są oni cenzurze i wypisywana jest ocenzurowana informacja o każdym pracowniku.

1.
  - a. Utwórz adnotację dla klasy która sygnalizuje, iż wszystkim polom typu String (zadeklarowanym lub dziedziczonym z nadklasy) należy ustawić wartość pustą
  - b. Ustaw tą adnotację na klasie Ksiegowca
2.
  - a. Stwórz adnotację dla pola, która sygnalizuje że jeśli pole jest typu String, to cenzor dokona podmiany tekstu na: pierwsze 3 znaki i "\_\_\_"  
Np. "Abrakadabra", otrzymamy: "Abr\_\_\_"
  - b. Ustaw tą adnotację na polach klas:  
"Inzynier" - "aktualnyProjekt"  
"Dyrektor" - "charakterystykaOsobowosci"
3.
  - a. Utwórz adnotację dla pola String, która pobiera tekst na podmianę i informuje iż zamiast tekstu w polu, które jest nią oznaczone wstawia tekst ustawiony jako parametr, a w przypadku jego braku, to używa do tego celu wykropkowanych ciągów, np. "Ala ma kota" -> "....."
  - b. Użyj tej adnotacji na polach klasy:  
"Dyrektor" - "miastoUrodzenia" – Tekst na podmianę: "Władystok"  
"Inzynier" – "opiniaKolegow" – brak tekstu do podmiany

4.
  - a. Stwórz adnotację, informującą, że pole o ile jest int-em ma mieć wartość 0
  - b. Zastosuj tą adnotację do pól klas:  
"Pracownik" - "wiek"  
"Monter" - "iloscWyprodukowanychElementow"
5.
  - a. Utwórz adnotację dla metody informującą, iż metoda ta ma być tyle razy wywołana, ile przekazemy w argumencie adnotacji
  - b. Zastosuj tą adnotację na metodzie "zwiększZysk" klas Monter (10), oraz Ksiegowo (3)
6.
  - a. Utwórz adnotację dla klasy, która pobiera parametr o dwóch możliwych wartościach (liczba lub Produkty – za pomocą enum) która sygnalizuje, że należy ustawić wartości wszystkich pól liczbowych na -1 lub pól z produktami na BOMBKI
  - b. Zastosować tą adnotację na klasie "Monter"
7.
  - a. Stwórz adnotację dla klasy informującą, iż obiekt danej klasy, jeżeli będzie miał wartość pola "wiek" < niż 30 zostanie usunięty z listy
  - b. Zastosować tą adnotację na klasie "Monter"