

Uwaga, przed rozpoczęciem rozwiązywania zadania należy zapoznać się z całością polecenia i wymagań.

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).

### Polecenie:

Przygotuj i przetestuj zestaw klas, interfejsów i metod:

1. Klasa Robot implementująca interfejs Porozumienie.
2. Klasa RobotDomowy dziedzicząca po klasie Robot, implementująca rozszerzoną wersję interfejsu Porozumienie o nazwie PorozumieniePlus.
3. Klasa RobotFabryczny dziedzicząca po klasie Robot.
4. Klasa RobotProdukcja dziedzicząca po klasie RobotFabryczny.
5. Klasa RobotSkladanie dziedziczy po RobotFabryczny, implementująca rozszerzoną wersję interfejsu Porozumienie o nazwie PorozumieniePlus.
6. Klasa SztucznaInteligencja.

Dobierz modyfikatory klas w taki sposób, aby:

- nie można było utworzyć żadnych obiektów klasy Robot ani RobotFabryczny – służą tylko do stworzenia hierarchii robotów
- nie można było dziedziczyć od klas RobotSkladanie oraz RobotProdukcja.

Pola składowe poszczególnych klas to:

1. Robot:
  - a. dataProdukcji – losowa wartość z przedziału lat (2000 - 2021) (wykorzystaj klasę LocalDate, oraz Random),
  - b. nazwa (łańcuch - jedno z dostępnych imion dla robotów), Dostępne imiona to: Adam1000, Roman2, Aleks900, Zosia Samosia, Ala2, Robert01, Ula,
  - c. identyfikator (losowo wygenerowany łańcuch 5 znaków, niezmienny podczas istnienia obiektu – wykorzystaj metodę do generowania łańcucha).
  - d. ileZadan (licznik informujący o tym, ile dany robot wykonał zadań od początku bieżącego cyklu, zmienna w momencie powstania nowego robota jest równa 0).
  - e. SztucznaInteligencja si (obiekt SztucznaInteligencja)
  - f. nrFabryczny – numer nadawany podczas tworzenia obiektu, niezmienny podczas istnienia obiektu (wykorzystaj pole statyczne oraz pole finalne)
2. RobotFabryczny:
  - a. zadanie (zmienna wyliczeniowa (enum) opisująca zadanie robota w fabryce, dostępne zadania: 1 – skrawanie, 2 – zgrzewanie, 3 – modelowanie, 4 – podnosnik, 5 – lakierowanie, 6 - sklejanie)
3. RobotSkladanie:
  - a. limit (ile w ciągu jednego cyklu produkcyjnego dany robot może wykonać zadań, zmienna losowana z przedziału [1,10])

## 4. Sztuczna Inteligencja:

- a. reprezentacja Danych – zmienna informująca o sposobie reprezentacji informacji, możliwe wartości to: „binarna”, „rozmyta”, „przybliżona”.

Ponadto, wyżej wymienione klasy posiadają następujące metody:

1. Konstruktory – parametryczne przyjmują
  - a. nazwa (w main nazwa przechowywana w tablicy i wybierana z niej dla robota)
  - b. zadanie – losowa wartość z enuma
2. odpowiednie gettery i setery – tylko niezbędne
3. toString – generuje string z informacją o robocie
4. equals – porównującą dwa dowolne obiekty robotów (które pola należy porównywać?)
5. zadanieWykonaj – metoda Robotów symulująca działanie robota, polega na zwiększeniu licznika ileZadan o losową liczbę z przedziału [1,20], w przypadku robotów z limitem - po przekroczeniu limitu metoda wypisuje na konsolę komunikat ostrzegający i nie zwiększa licznika ileZadan ponad wartość limit.

Przygotuj interfejsy dla wymienionych klas w taki sposób, aby:

1. Interfejs Porozumienie posiada metodę, której implementacja pozwoli każdemu robotowi wypisać na konsolę jak długo „żyje” (do teraz), a w przypadku robotów z limitem zadań, dodatkowo ile zadań jeszcze może wykonać. Nazwij tą metodę meldunek.
2. Interfejs PorozumieniePlus (dla RobotDomowy i RobotSkładanie) oprócz odziedziczonej metody meldunek posiada metodę powitanie, w której implementacja wypisuje na konsolę tekst „<nazwa> o numerze fabrycznym <nrFabryczny> mówi witaj”, np. „Zosia Samosia o numerze fabrycznym 3 mówi witaj”.

W funkcji main stwórz tablicę na Roboty i stwórz w niej: 3 obiekty RobotDomowy, 3 obiekty RobotProdukcja i 3 obiekty RobotSkładanie. W przypadku, gdy nie określono jawnie jak należy inicjować pole, wylosuj odpowiednią wartość.

W pętli:

- Wypisz na konsolę informacje o obiektach w Twoim programie.
- Przeprowadź symulację – zaprezentuj działanie wszystkich napisanych metod (poza getterami i setterami).
- Ponownie wypisz na konsolę informacje o obiektach w Twoim programie.

Sprawdź, czy w tablicy są dwa takie same roboty (za pomocą equals), jeśli tak wypisz w konsoli „Są identyczne roboty”, jeśli nie – „Brak identycznych robotów” – equals i sprawdzenie na dodatkowy 11 punkt.

Punktacja:

Uwagi:

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>• Przygotowanie klas – 5 pkt,</li><li>• Przygotowanie interfejsów: - 2 pkt</li><li>• Funkcja main – 3 pkt</li></ul> | <ul style="list-style-type: none"><li>• Kod musi być czytelny i staranny.</li><li>• Należy zadbać o dobry styl programowania (inkapsulacja, ..)</li></ul> |
|---|---|