GUI

Projekt nr 1

Z projektu można uzyskać maksymalnie 20 punktów. Termin oddania to 27 kwietnia

2023 r. godzina 18:00.

Klasa DzialPracownikow posiada pole nazwa. Nazwy działów muszą być unikalne.

Jeśli nazwa nie będzie unikalna, należy podnieść błąd “NotUniqueNameException“. Obiekt

klasy DzialPracowników można utworzyć tylko za pomocą metody “createDzial”, która

będzie przyjmowała odpowiednie argumenty. Dodatkowo powinna zostać utworzona metoda

aby uzyskać informację jacy pracownicy są w dziale.

Klasa abstrakcyjna Pracownik musi posiadać statyczną listę wszystkich utworzonych

pracowników, uzupełnianą o każdego nowo utworzonego pracownika. Klasa powinna

posiadać 4 pola i przyjmować ich wartości w konstruktorze: imię, nazwisko, datę urodzenia

oraz dział pracowników). Klasa powinna posiadać pole czyZdrowy o domyślnej wartości

true.W deklaracji klasy Pracownik dodaj interfejs Comparable i zaimplementuj metodę

compareTo, która będzie porównywała pracowników po wybranych przez autora kryteriach

(min. 2).

Klasa Trener dziedziczy po klasie Pracownik oraz posiada dodatkowe pole (String

specjalizacja).

Klasa Recepcjonista, dziedziczy po klasie Pracownik oraz posiada 2 dodatkowe pola:

String login, hasło - wartości tych pól powinny zostać podane w konstruktorze. Dodatkowo

posiada pole initial- tworzony z pierwszych liter imienia i nazwiska. Wartość initial powinna

być aktualizowana przy zmianach imienia lub nazwiska.

Klasa Manager, dziedziczy po klasie Recepcjonista. Każdy Manager powinien

posiadać do zwracania listy Zespołów w jakich brał udział oraz listę Zadań jakie obsługiwał

kiedy był w Zespole.

Klasa Zespol posiada 3 pola (String nazwa, Manager manager oraz listę

pracowników). Do zespołu można dodać pracowników za pomocą dwóch metod, jedna

metoda przyjmuje jednego pracownika, za to druga przyjmuje listę pracowników. Obie

metody mają mieć taką samą nazwę. Uwaga, nie powinno być możliwości dodania

managera jako szeregowego pracownika.

Klasa Zadanie ma dziedziczyć po klasie Thread oraz posiadać 6 pól (String nazwa,

String opis, stan enum(Utworzone,Rozpoczete, Zakonczone), dataUtworzenia,

dataZakończenia (należy skorzystać z klasy LocalDateTime) oraz czasWykonania - losowa

wartość pomiędzy 3 a 8 sekundami. Powinny być utworzone 2 konstruktory:

● Przyjmujący argumenty: Nazwa, opis, zatwierdzenie

● Przyjmujący argumeny: Nazwa

resztę pól proszę uzupełnić odpowiednimi domyślnymi wartościami.

Dostęp do pola stan powinno być ograniczone

Należy zaimplementować metodę “run()”, która rozpocznie wykonywanie zadania,

wyświetlając co sekundę odpowiednią informację.

/// kilka zadan / pojedyncze

Po zakończeniu Zadania powinna zostać ustawiona data Zakończenia na datę i czas chwili

zakończenia się Zadania (LocalDateTime.now()).

Dodatkowo należy zaimplementować metodę, która zwróci w jakim stanie jest Zadanie.

Funkcja powinna zwrócić jedną z 3 wartości: Utworzone, Rozpoczęte, Zakończone;

Klasa Praca ma implementować interface Runnable i posiadać następujące pola :

● kolekcje Zadań, na które ma czekać jak się wykonają, jeśli jest pusta może

się wykonać od razu

● String opis

● Zespol zespol

W klasie Praca powinna pojawić się statyczna metoda do pozyskania obiektu, podając ID

Zadania. Proszę do tego wykorzystać kolekcję typu - Map. Należy także zaimplementować

możliwość rozpoczęcia wykonywania wszystkich zadań. Uwaga, zadania oznaczone jako

“niezatwierdzone” należy ominąć, wyświetlając odpowiednią informację.

W projekcie ma być możliwość uzyskania informacji w jakich Zadaniach brał udział każdy

pracownik (sposób wykonania dowolny) oraz każdy dział ma mieć możliwość się dowiedzieć

się jakich posiada pracowników.

Uwagi ogólne:

Każdy obiekt powinien posiadać unikalny numer/identyfikator (increment), dla każdego

rodzaju klasy powinna być oddzielna numeracja.

Każda klasa powinna mieć zaimplementowaną metodę toString() oraz posiadać odpowiedni

konstruktor/y.

Jeżeli pole lub metoda wymagana (według autora) nie została opisana w powyższym opisie,

należy ją dodać zgodnie z własnym uznaniem.

Każda klasa dziedzicząca po klasie Pracownik powinna implementować interface

IDobryPracownik - interfejs ten powinien posiadać 3 metody wymyślone i zrealizowane

przez autora kodu.

Pola do których użytkownik nie powinien mieć dostępu powinny być ustawione jako private i

Posiadać gettery I settery

Dodatkowe zadania:

-Przy dodawaniu pracowników do Zespołu, można dodać obiektu klasy

Recepcjonista, ale nie można dodać obiekt klasy Manager.

-Zapis oraz odczyt danych przy wykonaniu każdej akcji do pliku