

# GUI

Projekt nr 1

12 marca 2022

Z projektu można uzyskać maksymalnie 20 punktów. Termin oddania to 07 maja 2022 r.

Klasa Dział Pracowników posiada pole "nazwa". Nazwy działów muszą być unikalne. Jeśli nazwa nie będzie unikalna, proszę wyrzucić błąd "NotUniqueNameException" (trzeba stworzyć taką klasę). Obiekt klasy Dział Pracowników można utworzyć tylko za pomocą metody "creatDzial", która będzie przyjmowała odpowiednie argumenty, aby można było utworzyć obiekt i go zwrócić w metodzie.

Klasa abstrakcyjna Pracownik musi posiadać statyczna listę wszystkich utworzonych pracowników. Konstruktor przyjmuje 4 pola (imię, nazwisko, data Urodzenia oraz dział pracowników). W deklaracji klasy Pracownika dodaj interfejs Comparable i zaimplementuj metodę compareTo, która będzie porównywała pracownika po imieniu, nazwisku i dacie urodzenia.

Klasa Kopacz, posiada jedno dodatkowe pole (narzędzie pracy - tekst).

Klasa Specjalista, posiada jedno dodatkowe pole (specjalizacja - tekst).

Klasa Użytkownik, przyjmuje do konstruktora dwa dodatkowe pole (login, hasło). Dodatkowo posiada pole inicjał- tworzony z pierwszych liter imienia i nazwiska, modyfikowany, kiedy się zmienia imię lub nazwisko.

Klasy Kopacz, Specjalista oraz Użytkownik dziedziczą po klasie Pracownik.

Klasa Brygadzysta dziedziczy po klasie Użytkownik. Każdy Brygadzysta posiada, zapisaną listę brygad, w których był.

Klasa Brygada posiada 3 pola (nazwa, brygadzystę oraz listę pracowników). Do brygady 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.

Klasa Zlecenie posiada numer zlecenia (numer jest generowany automatycznie, ale po kolei, czyli 1,2,3, ..), posiada listę Prac, brygadę, stan Zlecenia enum(Planowane, Nieplanowane, Realizowane, Zakończone), data Utworzenia, data Realizacji, data Zakończenia (proszę skorzystać z klasy LocalDateTime). Powinny być utworzone 4 konstruktory:

- z polem typu boolean (odpowiadający czy zlecenie jest planowany czy też nie)
- z polami typu boolean i typu brygada
- z polami typu boolean i typu lista prac
- z polami typu boolean, typu lista prac oraz brygada

, resztę pól proszę uzupełnić odpowiednimi domyślnymi wartościami.

Po utworzeniu obiektu jest możliwość dodania kolejnych prac poprzez metodę.

Powinna się pojawić w klasie możliwość dodania brygady jeśli taka nie została utworzona/dodana do zlecenia i powinna zwrócić true, w innym przypadku false.

Brygada raz dodana do Zlecenia, nie może być zmieniona.

Na koniec powinna pojawić się metoda “rozpocznijZlecenie”, która rozpocznie wykonywanie (jeśli posiada brygadę i lista prac nie jest pusta) Prac i ustawi datę Rozpoczęcia na teraz, prace muszą się wykonywać po kolei z listy.

Po zakończeniu Zlecenia powinna być ustawiona też data Zakończenia.

Klasa Praca ma dziedziczyć po klasie Thread i posiadać następujące pola:

- numer Pracy (generowany nowy numer dla każdej pracy)
- rodzaj Pracy - enum (Ogólna, Montaż, Demontaż, Wymiana)
- czas Pracy: int (w minutach)
- czy Zrealizowane: boolean
- opis: String

Konstruktor ma przyjmować rodzaj Pracy, czas oraz opis.

kiedy każda praca jest wykonywana ma wypisać na konsoli zawartość wszystkich swoich pól.

W klasie Praca oraz Zlecenie ma się pojawić statyczna metoda do pozyskania obiektu, podając numer pracy lub zlecenia. Proszę do tego wykorzystać kolekcję typu - Map.

W projekcie ma być możliwość uzyskania w jakich zleceniach brał udział brygadzista (sposób wykonania dowolny) oraz każdy dział ma mieć możliwość się dowiedzieć się jakich posiada pracowników.

Każda klasa musi mieć zaimplementowaną metodę toString(). Do projektu trzeba dodać przykład wykorzystania wszystkich metod, klas oraz konstruktorów klas.

Dodatkowe zadanie:

Przy dodawaniu pracowników do brygady, nie można dodać obiektu klasy Użytkownik. [1p.]