

Общи изисквания за двете задачи:

- Да се използват **статичен конструктор**
- Да се използват **статични методи**
- Да се прилагат **условни проверки** за валидация на данните
- Да се демонстрира работата на класовете в Main метода
- Да се обработват гранични случаи и грешни входни данни

Задача 1: Конвертор на температури

Условие:

Създайте клас `TemperatureConverter`, който да предоставя функционалност за конвертиране между градуси Целзий и Фаренхайт. Класът трябва да има следните характеристики:

Изисквания:

1. **Статичен конструктор**, който:
 - Инициализира стойността на абсолютната нула (-273.15°C)
 - Извежда съобщение за успешна инициализация на системата
2. **Статични методи** за конвертиране:
 - `CelsiusToFahrenheit(double celsius)` - конвертира от Целзий във Фаренхайт
 - `FahrenheitToCelsius(double fahrenheit)` - конвертира от Фаренхайт в Целзий
3. **Условия и валидации:**
 - Проверка дали входната температура не е под абсолютната нула
 - При невалидна температура да се извежда съобщение за грешка
 - Да се връща специална стойност (`double.MinValue`) при грешка
4. **Формули за конвертиране:**
 - $^{\circ}\text{F} = (^{\circ}\text{C} \times 9/5) + 32$
 - $^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times 5/9$

Тестови сценарии:

- Конвертиране на 25°C във Фаренхайт
- Конвертиране на 98.6°F в Целзий
- Опит за конвертиране на невалидна температура (-300°C)

Задача 2: Система за управление на потребители

Условие:

Създайте клас `UserManager` за управление на регистрацията на потребители в системата. Системата трябва да поддържа ограничен брой потребители и да валидира данните при регистрация.

Изисквания:

1. **Статичен конструктор**, който:
 - Инициализира брояча на потребители на 0
 - Задава максимален брой потребители (100)
 - Извежда информация за инициализиране на системата
2. **Статичен метод** `RegisterUser(string username, int age)`, КОЙТО:
 - Регистрира нов потребител при успешна валидация
 - Увеличава броя на потребителите при успешна регистрация
 - Връща `true` при успех и `false` при неуспех
3. **Статичен метод** `ShowStatistics()` за показване на:
 - Общ брой потребители
 - Максимален капацитет
 - Свободни места
 - Състояние на системата въз основа на натовареността
4. **Условия и валидации:**
 - Проверка за достигнат максимален брой потребители
 - Валидация на възрастта (13-120 години)
 - Проверка за празно или null потребителско име
 - Различни съобщения за състоянието според натовареността:
 - "ПЪЛНА КАПАЦИТЕТ" - при 100% запълненост
 - "ВИСОКО НАТОВАРВАНЕ" - над 80% запълненост
 - "НОРМАЛНО" - под 80% запълненост

Тестови сценарии:

- Регистриране на валидни потребители
- Опит за регистриране с празно име
- Опит за регистриране с невалидна възраст
- Показване на статистика след няколко регистрации