

Czas24h

Jesteś praktykantem w firmie informatycznej. Otrzymałeś zadanie zmodyfikowania klasy Czas24h napisanej w języku C#.

Obiekty klasy Czas24h reprezentują czas, t.j.: godzinę, minutę i sekundę w zapisie 24-godzinnym, w formacie h:mm:ss (np. 2:20:05). **Wewnętrznie czas jest zapamiętany jako liczba sekund, które upłynęły od godziny początkowej 0:00:00 - w prywatnym polu liczbaSekund** (patrz kod).

Twoim zadaniem jest zmodyfikować załączony kod klasy Czas24h tak, aby:

1. w konstruktorze nie dopuścić do utworzenia obiektu - zgłaszając wyjątek `ArgumentException` - w sytuacji przekazania niewłaściwych danych. Dopuszczalne wartości parametrów to: 0..59 dla sekund i minut, 0..23 dla godzin,
2. obiekty klasy Czas24h były zmiennicze (*mutable*) poprzez poprawne zdefiniowanie *properties* typu `set` dla godzin, minut i sekund (patrz kod).

Poniżej załączono kod, który będziesz modyfikować. Zawiera on zarówno klasę Czas24h jak i klasę Program z metodą Main(), która umożliwia testowanie zmian.

```
using System;
```

```
public class Program
```

```
{
    public static void Main()
    {
        string[] napis = null;
        Czas24h t = null;

        // wczytanie i parsowanie napisu oznaczającego godzinę, np. 2:15:27
        napis = Console.ReadLine().Split(':');
        int[] czas = Array.ConvertAll(napis, int.Parse);
        try
        {
            t = new Czas24h(czas[0], czas[1], czas[2]);
        }
        catch (ArgumentException)
        {
            Console.WriteLine("error");
            return;
        }

        // wczytanie liczby poleceń
        int liczbaPolecen = int.Parse(Console.ReadLine());

        for (int i = 1; i <= liczbaPolecen; i++)
        {
            // wczytanie polecenia
            napis = Console.ReadLine().Split(' ');
            string typPolecenia = napis[0];
            int liczba = int.Parse(napis[1]);

            // wykonanie polecenia na obiekcie t typu Czas24h
            try
            {
```

```

        switch (typPolecenia)
        {
            case "g":
                t.Godzina = liczba;
                break;
            case "m":
                t.Minuta = liczba;
                break;
            case "s":
                t.Sekunda = liczba;
                break;
        }
    }
    catch (ArgumentException)
    {
        Console.WriteLine("error");
        return;
    }
}
Console.WriteLine(t);
}
}

public class Czas24h
{
    private int liczbaSekund;

    public int Sekunda
    {
        get => liczbaSekund - Godzina * 60 * 60 - Minuta * 60;
        // uzupełnij kod - zdefiniuj setters'a
    }

    public int Minuta
    {
        get => (liczbaSekund / 60) % 60;
        // uzupełnij kod - zdefiniuj setters'a
    }

    public int Godzina
    {
        get => liczbaSekund / 3600;
        // uzupełnij kod - zdefiniuj setters'a
    }

    public Czas24h(int godzina, int minuta, int sekunda)
    {
        // uzupełnij kod zgłaszając wyjątek ArgumentException
        // w sytuacji niepoprawnych danych

        liczbaSekund = sekunda + 60 * minuta + 3600 * godzina;
    }

    public override string ToString() => $"{Godzina}:{Minuta:D2}:{Sekunda:D2}";
}

```

Polecenie

1. Skopiuj kod i uzupełnij go zgodnie z podanymi powyżej warunkami. Nie wolno

modyfikować istniejącego kodu - jedynie uzupełniasz brakujące fragmenty (wskazane komentarzem).

2. Prześlij całość (kod klasy Program z metodą Main() oraz kod klasy Czas24h) do oceny przez sędziego.

Weryfikacja sędziego

Sędzia uruchomi przesłany przez Ciebie kod, dostarczając mu na standardowe wejście dane testowe w ustalonym formacie i kolejności oraz weryfikując poprawność wypisywanych wyników.

Specyfikacja danych wejściowych

1. W pierwszej linii napis składający się z trzech liczb całkowitych oddzielonych dwukropkiem (np. 2:20:15) - opisujący czas początkowy.
2. W drugiej linii liczba n poleceń modyfikujących - liczba całkowita w zakresie od 1 do 100 włącznie (np. 3).
3. W kolejnych n liniach polecenia trzech różnych typów - g, m, s - zmieniające czas:
 - polecenie g 1 zmienia godzinę na 1
 - polecenie m 20 zmienia minutę na 20
 - polecenie s 15 zmienia sekundę na 15

Polecenia danego typu mogą wystąpić wielokrotnie lub nie wystąpić wcale.

Format polecenia: znak spacja liczba całkowita

Specyfikacja wypisywanego wyniku

- W przypadku wczytania niepoprawnych danych (przy tworzeniu obiektu typu Czas24h lub przy jego modyfikacji), program wypisuje tekst error.
- W przypadku poprawnych danych program wypisuje aktualny czas po wykonaniu wszystkich poleceń, w formacie h:mm:ss.

Przykłady

Przykład 1

Wejście:

2:15:37

3

m 20

g 1

s 26

Wyjście:

1:20:26

Przykład 2

Wejście:

2:15:37

2

m 20

g 24

Wyjście:

error

Przykład 3

Wejście:

2:15:37

2

m -20

g 23

Wyjście:

error

Przykład 4

Wejście:

2:15:37

8

m 20

g 23

g 1

s 15

m 10

s 23

s 1

m 12

Wyjście:

1:12:01

Przykład 5

Wejście:

24:15:37

2

m 20

g 1

Wyjście:

error

Weryfikowane umiejętności

- czytanie/analizowanie kodu
- arytmetyka modulo
- konwersje typów, rzutowanie

Ocenianie

Ocena końcowa jest % liczby poprawnie zaliczonych testów.