

Упражнения: Дефиниране на класове 2

Забележка: За задачите с дробни да се спазва общото ограничение, че знаменателя винаги е естествено число. Т.е. не може да е 0 или отрицателно число.

Предложение за модификация: Ако ограничението за знаменателя се сведе до **цяло число**, то решете задачите, като обработвате случаите с въведен знаменател **0** и тогава да се извежда съобщение за невалидни входни данни, а за отрицателна стойност на знаменателя – да се промени метода за подреждане на изходната колекция

Problem 1. Дефиниране на клас Рационално число

Дефинирайте клас **RacionalNumber** с **private** полета **numerator**(числител) и **denominator**(знаменател).

Създайте 3 обекта и ги изведете във формат "Numerator/denominator {numerator}/{denumaerator}":

Примери

Вход	Изход
3	3/4
4	5/6
5	7/8
6	
7	
8	

Problem 2. Дефиниране на клас Рационално число *

Дефинирайте клас **RacionalNumber** с **private** полета **numerator**(числител) и **denominator**(знаменател).

Въведете ги на един ред, като за разделител ползвате " " и ги изведете във формат "Numerator/denominator {numerator}/{denumaerator}", на един ред, разделени със "; "

Примери

Вход	Изход
3 4 5 6 7 8	3/4; 5/6; 7/8

Problem 3. Несъкратима дроб *

Дефинирайте клас **RacionalNumber** с **private** полета **numerator**(числител) и **denominator**(знаменател). Въведете от клавиатурата числа на един ред, разделени с интервал които да бъдат числители и знаменатели на дробите (както в предишния пример) и ги запишете в списък от такива **RacionalNumber** обекти. Създайте нов списък, в който са преобразувани така въведените дроби в несъкратими и изведете новия списък.

Подсказки

Създайте метод **int BiggestDivider(int numerator, int denominator)**, който намира Най-големия общ делител на числителя и знаменателя **НОД(числител и знаменател)**. Напишете конструктор, който приема числител и знаменател, извиква в себе си метода **int BiggestDivider(int numerator, int denominator)** и записва в новия списък съкратената дроб по следното правило:

`nod = int BiggestDivider(int numerator, int denominator) => {numerator/nod} {denominator/nod}`. Ползвайте известни алгоритми за намиране на НОД (на Евклид и други)

Примери

Вход	Изход
3 4 3 6 25 100	3/4; 1/2; 1/4

Бонус

1. Изведете новия списък, подреден във възходящ ред
2. Ако не се спазва ограничението за отрицателни знаменатели, вижте дали се налага модификация на метода за подреждане във възходящ или низходящ ред
3. Да се модифицира така, че посоката на подреждане да се въвежда от клавиатурата
4. Задачата да се реши с функционално програмиране

Problem 4. Несъкратима дроб **

Решете задачата, с функционално програмиране, като за целта използвате:

1. Функция, която намира НОД(числител, знаменател)
2. Функция, която преобразува дроб, в несъкратима, чрез деление на числителя и знаменателя на НОД-а им.

Problem 5. Дефиниране на клас Четно число *

Дефинирайте клас **EvenNumber** с **private** поле **num**(числител). Който да ползвате в следната задача:

От клавиатурата на един ред се въвеждат няколко числа и само четните се извеждат. За целта да се ползва списък от числа, в които да се запомнят всички въведени числа. Тези, които са четни от тях да се запаметят в клас **EvenNumber** и да се изведат на един ред, с разделител запетая.

Примери

Вход	Изход
3 4 5 6 7 8	4, 6, 8

Problem 6. Дефиниране на клас Четно число **

Задачата да се реши като се ползва функционално програмиране и ламбда израз за проверка на четността на числата.

Problem 7. Дефиниране на клас Нечетно число

Решете предишната задача, с условие, че извеждате нечетни числа.

Problem 8. Дефиниране на клас Кратно на „к“ число

Решете задача 5, с условие, че извеждате числата, кратни на **число k**, което се въвежда от клавиатурата

- На **Първи ред** се въвежда последователност от числа, разделени с интервал
- На **втори ред** се въвежда число **k** – на което да са кратни
- На **конзолата** се извеждат кратните на **k**, разделени със запетая

Примери

Вход	Изход	Вход	Изход
3 4 5 6 7 8 9 15 28 3	3,6,9,15	3 4 5 6 7 8 9 15 28 4	4,8,28

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма **"Обучение за ИТ кариера"** на МОН за подготовка по професия "Приложен програмист".



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).



SoftUni
Foundation

