Project Description

Saturday, January 20, 2018

5:55 PM

Clipped from: <https://courses.edx.org/courses/course-v1:Microsoft+DEV277x+1T2018/courseware/b8fd0737466e4e7a80aea3d469f9c7a7/5d23269aaa1d40d69da0f0478e7ea57d/?child=first>

Microsoft: DEV277X 
Object Oriented Programming in Java 
Help 
bhansalirushabh 
Course 
b is 
b is 
b is 
Discussion 
Wiki 
Progress 
Course > Module 2 Building Your Own Objects > Project- FracCalc > Project Description 
< Previous 
Next > 
scription 
Module 2 Project - Fraction 
his project is designed to help you practice building your own object class and testing it with a client class. You will be creating two 
lasses, one called Fraction and the other called FractionCalculator. The Fraction class is an object that holds information about a fraction 
(numerator and denominator). It will have several constructors and both private and public methods implementing the behavior of a 
raction. The FractionCalculator class is a class that will allow the user to enter in fractions and operations, calculating and displaying the 
result. It will run until the user tells it to quit. When this program is complete, you wont have to second guess your fraction arithmetic 
er again! 
Part 1 - Fraction Class 
Create a new class called "Fraction" and include the following: 
Fields 
two private instance variables to hold the numerator and denominator as ints 
nstructors 
a two parameter constructor that initializes the numerator and denominator 
• This constructor should throw an IllegalArgumentException if the denominator is zero 
• If the user enters a negative denominator bump the negative sign to the numerator. For example, -3/-2 should be converted to 3/2. 
Likewise, 5/-3 should be converted to -5/3 
• example: Fraction myFrac = new Fraction(4,5); creates a fraction who's numerator is 4 and denominator is 5 
one parameter constructor that initializes the object equal in value to the integer parameter. 
• example: Fraction myFrac = new Fraction(3) would create a Fraction with numerator equal to 3 and denominator equal to 1. 
zero parameter constructor that initializes the object to 0, meaning the numerator is 0 and the denominator is 1 
• example: Fraction myFrac = new Fraction(); would create a fraction who's decimal value is 0 
ou should eliminate as much redundancy as possible by letting your constructors rely on one another using the "this" keyword. 
Methods 
Method to 
implement 
getNumerator() 
getDenominatorO 
tostring() 
toDouble() 
add() 
subtract() 
muttipty() 
divide() 
equals() 
toLowestTerms() 
gcd() 
uals() 
paramete 
none 
none 
Fraction 
other 
Fraction 
other 
Fraction 
other 
Fraction 
other 
Object 
other 
none 
int num, int 
den 
retu 
Strin 
doubl 
Fracti 
Fracti 
Fracti 
Fracti 
boole 
none 
description 
exposes the value of the numerator field to the user 
exposes the value of the denominator field to the user 
"numerator/denominator", a String representation of the Fraction 
the result of numerator / denominator 
returns a new Fraction that is the sum of other and this fractions 
returns a new Fraction that is the difference beüeen the other and this fraction 
returns a new Fraction that is the product of the other and this fraction 
returns a new Fraction that is the division of the other and this fraction, throw an IllegalArgumentException() if 
the user asks you to divide by O 
must take in an "Object" to property override the Object class's equals method, but should ultimately check if 
tvwo fractions are equal 
converts the current fraction to the lowest terms 
takes in tv•vo ints and determines the greatest common divisor of the tv'/0 ints, should be a static method 
Override the Object equals() method so that it accurately determines whether or not two fractions are equal. In order to have it override, i 
has to take an Object as a parameter. Your method should check whether or not the parameter is an instanceof Fraction, since if it is not a 
Fraction it cannot be equal. Dont forget to cast the parameter to a Fraction after you check if it is an Object of type Fraction so that you 
n access its variables. Two fractions are equal if they represent the same number (i.e. 3/6 = 1/2 and -2/3 = 2/-3). 
oLowestTerms() 
o convert a fraction to lowest terms we have to determine the greatest common divisor (factor) between the numerator and 
denominator. The greatest common divisor of two numbers a and b, is the largest number that evenly divides both a and b. 
he Euclidean Algorithm is a fast method for determining the GCD of two numbers. Here is pseudocode for its implementation: 
hi le a and b are not zero 
find the remainder of a divided by b 
set a to b 
set b to the remainder you found 
return a 
Here is an example of how it would work if a is 105 and b is 147. 
Loop Iteration 
Pre-Loop 
2 
3 
4 
cd() 
Value of a at end of loop run 
105 
147 
105 
42 
21 
Value of b at end of loop run 
147 
105 
42 
21 
105 % 147 - 105 
147 % 105 - 147 - 105 = 42 
105 % 42 = 105 -84 = 21 
21 = o 
mplement gcd() as a public static method that takes two integers as parameters and returns an int that is their greatest common divisor. 
Part 2 — FractionCalculator Class 
n this section, you will implement a FractionCalculator class that has a main method and three helper methods. Here is a screenshot from 
sample run: 
This program 
It will add, 
Please enter 
Please enter 
Please enter 
Please enter 
1/2 + 1/3 
is a fraction calculator 
subtract, multiply and divide fractions until you type Q to quit. 
your fractions in the form a/ b, where a and b are integers. 
Please 
Please 
Please 
Please 
Please 
Please 
3/12 
Please 
Please 
Please 
Please 
Please 
Please 
50/100 
Please 
ente r 
enter 
enter 
enter 
ente r 
ente r 
enter 
ente r 
ente r 
ente r 
ente r 
enter 
= 1/2 
enter 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
is true 
an operation (+, 
or 
or 
or 
or 
or 
or 
or 
or 
or 
or 
— or Q 
integer 
integer 
— or Q 
intege r 
integer 
— or Q 
integer 
integer 
— or Q 
intege r 
integer 
— or Q 
integer 
integer 
— or Q 
to quit) : 
1/2 
to quit) : 
to quit) : 
3/12 
to quit) : 
to quit): — 
50/100 
to quit): 
Process finished with exit code O 
our program should be robust so that it if the user enters invalid input it will continue to re-prompt them until it is valid. Here is an 
ample run where the user is confused and enters invalid input: 
'Library/Java/JavaVirtua1Machines/jdk1.8.ø_65.jdk/Contents/Home/bin/java . 
This program is a fraction calculator 
It wilt add, subtract, multiply and divide fractions until you 
Please enter your fractions in the form alb, where a and b are 
Please enter an operation (4, — 
= or Q to quit): foo 
type Q to quit. 
integers. 
Invalid input (4, 
Invalid input (4, 
Invalid input (4, 
Invalid input (4, / * 
Invalid input (4, 
or Q to quit): operation 
/ , = or Q to quit): 1/2 
/ , = or Q to quit): +— 
— or Q to quit): / * 
— or Q to quit): / 
Please enter a fraction (a/b) or integer (a): 
th ree 
Invalid fraction. Please enter (alb) or (a), 
wheæ a 
Invalid fraction. Please enter (alb) or (a), 
where a 
Invalid fraction. Please enter (alb) or (a), 
wheæ a 
Invalid fraction. Please enter (alb) or (a), 
where a 
Invalid fraction. Please enter (alb) or (a), 
where a 
Please enter a fraction (alb) or integer (a): ø/2 
and 
and 
and 
and 
and 
b are 
b are 
b are 
b are 
b are 
integers 
integers 
integers 
integers 
integers 
and 
and 
and 
and 
and 
not 
not 
not 
not 
not 
zero: 
zero: 
zero: 
zero: 
zero: 
one/2 
1/3 
1/3 / = undefined 
Please enter an operation (4, — 
Process finished with exit code 
Methods 
Method to implement 
/ , = or Q to quit): q 
param 
eter 
description 

Fr

"q", or "Q" it should re-prompt 
them until there is valid input. 
valid Fraction() 
getFraction() 
etOperation() 
getoperation() 
b is 
b is 
b is 
Scanne 
r input 
String 
input 
Scanne 
r input 
Stri 
ng 
bo 
ole 
Fra 
ctio 
Asks the user to enter in a valid mathematical operation. If the user enters anything 
except " 
returns true if the parameter is in the form "alb" where a is any int and b is any positive 
int 
It prompts the user for a String that is a vafidFraction. If they enter any thing that is not 
a valid Fraction, it should re-prompt them until it is valid 
Here is example output from a call to getOperation(): 
Please enter a fraction (alb) or integer (a): 
fraction! 
Invalid fraction. Please 
Invalid fraction. Please 
Invalid fraction. Please 
Invalid fraction. Please 
Invalid fraction. Please 
Invalid fraction. Please 
Invalid fraction. Please 
ould have returned " *". 
alidFraction() 
enter 
enter 
enter 
enter 
enter 
enter 
enter 
(alb) 
(a/b) 
(alb) 
(alb) 
(alb) 
(a/b) 
(a/b) 
or 
or 
or 
or 
or 
or 
or 
(a), 
(a), 
(a), 
(a), 
(a), 
(a), 
(a), 
where 
whe re 
where 
where 
whe re 
whe re 
whe re 
and 
a 
a and 
and 
a 
and 
a 
and 
a 
and 
a 
and 
a 
b are 
b are 
b are 
b are 
b are 
b are 
b are 
integers 
integers 
integers 
integers 
integers 
integers 
integers 
and 
and 
and 
and 
and 
and 
and 
not 
not 
not 
not 
not 
not 
not 
th ree 
zero: 
two/ one 
zern: 
3/ two 
zero: 
At the end of this run, getOperation 
1/2 
zero: 
ø.5 
zero: 
2/-1 
zero: 
: -31 
me things to be mindful of when implementing the validFraction() method: 
The first character may or may not be a "-n character. If a negative shows up anywhere else, then it is not a valid fraction. It may be 
helpful to remove the character if there is one. 
If there is no "/" character, then every character in the string must be a number (if you removed the "-" sign). 
If there is a character, then it may be helpful to create substrings for the numerator and denominator. 
• Both substrings must be non-empty. 
• Both must be entirely made of numbers. 
• The denominator cannot be "0". 
Hint 1: It may be useful to create a helper method isNumber() that takes a String as input and returns true if every character in the String 
is a number 0-9 and false otherwise. This method can also check for empty strings. Hint 2: Once you determine whether or not the 
trings are numbers, you may find the Integer.parselnt() method helpful. 
etFraction() 
Here is example output from a call to getFraction(). If the user enters any thing that is not a valid Fraction, it should re-prompt them 
Please enter a fraction (alb) or integer (a): 
Invalid fraction. Please enter (alb) or (a), 
Invalid fraction. Please enter (a/b) or (a), 
until it is valid: Invalid fraction. Please enter (alb) or (a) , 
Invalid fraction. Please enter (alb) or (a), 
Invalid fraction. Please enter (alb) or (a), 
Invalid fraction. Please enter (a/b) or (a), 
Invalid fraction. Please enter (a/b) or (a), 
fraction ! 
where a and b are integers and b is not zero: three 
where a and b are integers and b is not zero: two/one 
where a and b are integers and b is not zerw. 3/ two 
wheæ a and b are integers and b is not zero: 1 / 2 
where a and b are integers and b is not zern: ø.5 
where a and b are integers and b is not zero: 2/—1 
where a and b are integers and b is not zero: —31 
This call would return a 
new Fraction object equal to -31/1. No user input should throw an exception! If you are getting exceptions, then it is likely your 
alidFraction method isnt correct. 
Part 3 - Putting it all together! 
. Write a short introduction method that describes the calculator program and welcomes your user 
. Ask the user to enter in an operation 
. As long as the user enters something that's not "q" or "Q" when asked for an operation you should run the calculator 
Get two fractions from the user and then perform whichever operation they ask for 
. Print the result of the operation 
Here is an example run of the entire program: 
This program 
It will add, 
Please enter 
Please enter 
Please enter 
Please enter 
1/2 + 1/3 
is a fraction calculator 
subtract, multiply and divide fractions until you type Q to quit. 
your fractions In the form a/ b, where a and b are integers. 
Please 
Please 
Please 
Please 
Please 
Please 
3/12 
Please 
Please 
Please 
Please 
Please 
Please 
50/100 
Please 
ente r 
ente r 
enter 
ente r 
ente r 
enter 
enter 
enter 
ente r 
enter 
ente r 
enter 
= 1/2 
enter 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
an operation (+, 
a fraction (a/ b) 
a fraction (a/ b) 
is true 
an operation (+, 
or 
or 
or 
or 
or 
or 
or 
or 
or 
— or Q to quit) : 
intege r 
intege r 
— or Q to quit) : 
intege r 
intege r 
— or Q to quit) : 
integer 
3/12 
integer 
— or Q to quit) : 
intege r 
intege r 
= or Q to quit): — 
intege r 
50/100 
intege r 
1/2 
— or Q to quit): 
Process finished with exit code O 
Part 4 - Hacker Problem - FractionCalculatorAdvanced 
reate another class called FractionCalculatorAdvanced. You may cut and paste useful methods from FractionCalculator. The key differenc 
between FractionCalculator and FractionCalculatorAdvanced is that the user can enter in their operations on a single line. Allow the user to 
nter their input onto a single line. Your program must be robust so that if the user enters in invalid input, it will re-prompt them until 
hey either enter a q to quit or a valid operation. It is possible to do this without try'/catch, but it is quite difficult. You may read about 
ry'/catch blocks here: http://beginnersbook.com/2013/04/try-catch-in-java/. Here is sample output from a run of 
/Library /Java/JavaVirtuatMachines/j dkl. 8. Ø_65. j dk/Contents/Home/bin/j ava 
This program is a fraction calculator 
It wilt add, subtract, multiply and divide fractions until you type Q to quit. 
Valid operations are of the form " [FRACI [OPERATICNI [FRACI". 
[FRACI can be either a single integer or two integers separated by 't/". 
[OPERATIONI can be 
Enter an operation 
Invalid operation. 
Enter an operation 
Invalid operation. 
Enter an operation 
Invalid operation. 
Enter an operation 
Invalid operation. 
FractionCalculatorAdvanced: Enter an operation 
Invalid operation. 
Enter an operation 
Invalid operation. 
Enter an operation 
45 / 22 = 45/22 
Enter an operation 
33 / 11 - 
Enter an operation 
1/2 * 4/3 = 2/3 
Enter an operation 
, / or — 
(q to quit): this 
Must be " [FRAC] [OPERATION] 
(q to quit): isn't 
Must be " [FRAC] [OPERATION] 
(q to quit): a vat id 
Must be " [FRAC] [OPERATION] 
(q to quit): op + ation 
Must be " [FRACI [OPERATION] 
(q to quit): 2 + foo 
Must be " [FRAC] [OPERATION] 
(q to quit): 3 — zee 
Must be " [FRAC] [OPERATION] 
[FRAC] " • 
[FRAC] 't • 
[FRAC] " • 
[FRAC] • 
[FMC] • 
[FRAC] 't • 
(q to quit): 
(q to quit): 
(q to quit): 
(q to quit): 
45 / 22 
33 / 11 
1/2 * 4/3 
11/14 / 7/11 
11/14 / 7/11 - 
121/98 
Enter an operation (q to quit): 
Process finished with exit code e 
< Previous 
Next > 
Learn About Verified Certificates 
About edX for dusiness Blog News Help Center Contact 
Terms of Service & Honor Code Privacy Policy Accessibility Policy 
Careers 
Sitemap 
Donate 
Media Kit 
@ All Rights Reserved 
noooea 
Download on the 
App Store 
ANOROIO APP ON 
Google play 
0 2012—2018 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX 
and Open edX logos are registered trademarks or trademarks of edX Inc. I Elcpfi17044299 
OPFNIZX 
Support 