



Задачи по предмету «Геометрические задачи вариационного исчисления»

(на самом деле C#)

Решение присылать в виде полной структуры проекта с .sln и .csproj файлами

У каждой задачи есть минимальный набор требований, реализация дополнительных  требований оценивается дополнительными баллами, но лишних  телодвижений делать не следует


Делать все сразу не стоит – присылайте частями, так как части может уже хватить до зачета.


Реализовывать нужно на C# а не на Java с синтаксисом C# - поэтому вспоминайте, например, что существуют свойства (properties) и инициализаторы коллекций (collection initializers)

Оранжевые задачи по двойному тарифу

#1 Rational Numbers


Требуется реализовать структуру (**struct**), содержащую представление рационального числа в виде целой части, числителя и знаменателя. Для нее предоставить перегрузки арифметических операций и конвертацию в типы **double** и **decimal**


 Представление в виде неправильной дроби, преобразование неправильной дроби в смешанную

 Преобразование из **double** и **decimal**, Сокращение дроби, приведение к общему знаменателю

#2 Money


Требуется реализовать два типа (**type**). Первый - структура **Money** (**struct**) – для описания денежной суммы в одной валюте, должна содержать в себе численное значение суммы в виде **decimal** поля и какого-то идентификатора валюты. Предоставить перегрузки арифметики (сложение и вычитание с **Money** с той же валютой, умножение и деление на **decimal**). Второй – класс **Payment** (**class**), содержащий в себе коллекцию **Money** с уникальными валютами. Для него перегрузить арифметику (сложение и вычитание, с сохранением уникальности валюты)


 Поддержка идентификации валют по ISO 4217 (http://en.wikipedia.org/wiki/ISO_4217), перегрузка **ToString()** с сохранением информацией о валюте.

 Сложные системы конвертации валют, соединения с сервисами cbr.ru, управляемое форматирование

#3 Line Counting

Требуется создать программу, которая принимает на вход путь к папке, находит в ней все файлы с расширением .cs и подсчитывает в них количество строк кода. Пустые линии должны игнорироваться.

 Поддержка рекурсивного просмотра каталога, игнорирование строк комментариев

 Анализ многострочных литералов, определение конца строки по

#4 Poor man's Expression Tree

Требуется реализовать примитивное бинарное дерево вычисления выражения. Дерево должно состоять из промежуточных узлов и листьев. Лист дерева должен иметь тип `Leaf<T,V>`, свойства `T Value` и `Func<T,V> Converter`, то есть значение типа `T` и функцию конвертации этого значения в тип `V`. Промежуточный узел должен иметь тип `Node<V>` и свойство `Func<V,V,V> Operation`, то есть операцию позволяющую взять два значения `V` из поддеревьев и выдать значение типа `V`. У самого дерева должен быть метод `V Calculate()`, позволяющий собрать значения с листьев и выдать результат вычисленного выражения.

- Более обобщенный тип дерева, с промежуточными узлами типа `Node<L,R,V>`, со свойством `Func<in L, in R, out V> Operation`, то есть с возможностью комбинировать значения разных типов из поддеревьев и выдавать значение третьего типа.
- Автобалансировка, упрощения, неполные узлы, и вообще универсализация до полноценного дерева выражений

#5 Emit Calculator

Требуется реализовать примитивный генератор калькулятора. На вход подается в текстовом виде арифметическое выражение, содержащее имена переменных `a,b,c,d` и знаки операций `+` и `*`, в обратной польской записи. Требуется по этому выражению построить динамически IL-код, вычисляющий данное выражение, и запаковать данный код в делегат `Func<int,int,int,int,int>` (с помощью `System.Reflection.Emit.DynamicMethod` и его метода `GetILGenerator()`), далее, с помощью данного делегата вычислить значение выражения для конкретных значений `a,b,c` и `d`

- Сделать перегрузку также и для `Func<double,double,double,double,double>`, парсить выражение из обычной, а не обратной польской записи
- Любая универсализация: неопределенное число переменных, константы, скобки

#6 Poor man's Shutdown

Требуется реализовать примитивный аналог утилиты shutdown (только поддержка флагов `-l`, `-r`, `-s` и `-t` <https://technet.microsoft.com/en-us/library/bb491003.aspx>) Требуется распарсить аргументы командной строки, выдержать затребованный таймаут и выполнить действие. В случае если имеется возможность (Windows) – выполнить вызов функции `ExitWindowsEx`, экспортированной из `user32.dll` с соответствующим набором параметров ([https://msdn.microsoft.com/en-us/library/windows/desktop/aa376868\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa376868(v=vs.85).aspx)). В ином случае – вместо вызова использовать заглушку, выводящую параметры вызова в консоль, или аналогичный функционал на целевой оси (`/usr/bin/shutdown`)

-
- Поддержка сетевого доступа и прочие тонкости.