

УТВЕРЖДЕН

Заведующий кафедрой «Управление
разработкой программного обеспечения»
_____ / Авдошин С.М./
« ____ » _____ 2011 г.

**ПРОГРАММА ВИЗУАЛИЗАЦИИ ОПЕРАЦИЙ НАД
КВАТЕРНИОНАМИ НА ПЛАТФОРМЕ WINDOWS PHONE 7**

Руководство программиста

ЛИСТ УТВЕРЖДЕНИЯ

Инв. № подп.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Руководитель работы

_____ / Гринкруг Е.М./
« ____ » _____ 2011 г.

Исполнитель: студент группы 171ПИ

_____ / Дубов М.С. /
« ____ » _____ 2011 г.

Национальный исследовательский университет – Высшая школа экономики
Факультет бизнес-информатики, отделение программной инженерии

УТВЕРЖДЕН

**ПРОГРАММА ВИЗУАЛИЗАЦИИ ОПЕРАЦИЙ НАД
КВАТЕРНИОНАМИ НА ПЛАТФОРМЕ WINDOWS PHONE 7**

Руководство программиста

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Листов 17

Содержание

1. Назначение и условия применения программы	3
1.1. Функциональное назначение	3
1.2. Эксплуатационное назначение	3
1.3. Требования к программной совместимости	3
1.4. Требования к составу и параметрам технических средств	3
2. Характеристики программы	3
2.1. Временные характеристики	3
2.2. Характеристики памяти	3
2.3. Режим работы	4
3. Обращение к программе	4
3.1. Библиотека DCL.Maths	4
3.1.1. Структура Quaternion	4
3.1.2. Структура Fraction	5
3.1.3. Структура Angle	6
3.1.4. Статический класс Common	7
3.2. Библиотека DCL.Phone.Xna	9
3.2.1. Абстрактный класс Shape	9
3.2.2. Класс Ellipse – наследник Shape	10
3.2.3. Класс Circle – наследник Ellipse	11
3.2.4. Класс Ellipsoid – наследник Shape	11
3.2.5. Класс Sphere – наследник Ellipse	12
3.2.6. Класс Dot – наследник Shape	12
3.2.7. Перечисление DrawMode	13
3.2.8. Класс PivotGame – наследник Microsoft.Xna.Framework.Game	13
3.2.9. Класс PivotGameItem	15
3.2.10. Класс SelectionChangedEventArgs – наследник EventArgs	15
3.2.11. Структура DrawingArea	15
3.2.12. Перечисление ProjectionType	16
4. Входные и выходные данные	16
4.1. Организация входной информации	16
4.2. Организация выходной информации	16
5. Сообщения	16

1. Назначение и условия применения программы

1.1. Функциональное назначение

Библиотека DCL.Maths реализует основные операции над кватернионами, обыкновенными дробями и градусами, а также предоставляет некоторый общий функционал, расширяющий возможности классов .Net Framework.

Библиотека DCL.Phone.Xna содержит классы, отвечающие за построение и преобразование трехмерных моделей посредством кватернионов, а также средства построения пользовательских интерфейсов на платформе Microsoft® Xna Framework.

1.2. Эксплуатационное назначение

Библиотека DCL.Maths предназначена для использования в составе программных продуктов, реализация которых задействует предоставляемый библиотекой функционал.

Библиотека DCL.Phone.Xna предназначена для использования в составе программных продуктов для платформы Microsoft® Windows® Phone 7, разрабатываемых с использованием Microsoft® Xna Framework.

1.3. Требования к программной совместимости

Библиотека DCL.Maths требует для своей работы среду, предоставляющую платформу Microsoft®.Net Framework.

Библиотека DCL.Phone.Xna требует для своей работы среду Xna Framework for Windows Phone, которая входит в состав системы Microsoft® Windows® Phone 7.

1.4. Требования к составу и параметрам технических средств

Библиотека распространяется на оптических дисках и через интернет. Для чтения оптических дисков компьютер должен быть оснащен устройством чтения дисков CD; операционная система должна поддерживать файловую систему iso9660.

2. Характеристики программы

2.1. Временные характеристики

Время построения трехмерных моделей, а также время выполнения основных операций над ними (поворот с помощью кватернионов, масштабирование) находится в квадратичной зависимости от параметра, задающего детализацию модели.

Операция перемножения двух кватернионов требует 16 умножений и 12 сложений. Операция умножения кватерниона на чисто векторный кватернион оптимизирована и требует 12 умножений и 8 сложений.

2.2. Характеристики памяти

Память, необходимая для хранения массива вершин каждой трехмерной модели, находится в квадратичной зависимости от параметра детализации.

Объект, представляющий кватернион, хранится в памяти как совокупность четырех вещественных чисел одинарной точности (типа *System.Single*). Объекты, представляющие обыкновенную дробь и градусную величину, хранятся как совокупности двух и трех целочисленных значений соответственно.

2.3. Режим работы

Конструкторы классов и структур, на основе которых создаются объекты, представляющие кватернионы, дроби, углы и трехмерные модели, определяют ошибки во входных данных, и в случае обнаружения таковых прекращают обработку данных. Восстановление программы после сбоя не предусмотрено.

3. Обращение к программе

3.1. Библиотека *DCL.Maths*

3.1.1. Структура *Quaternion*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
Re	public	Single		Вещественная часть
X	public	Single		Коэффициент при i
Y	public	Single		Коэффициент при j
Z	public	Single		Коэффициент при k
i	public static readonly	Quaternion		Мнимая единица i
j	public static readonly	Quaternion		Мнимая единица j
k	public static readonly	Quaternion		Мнимая единица k
Свойства				
VectorPart	public	Vector3		Векторная часть
Abs	public	Single		Модуль
Конструкторы				
Quaternion	public		Single	Конструктор приведения типов
Quaternion	public		Vector3	Конструктор приведения типов
Quaternion	public		Single, Single, Single, Single	Конструктор общего вида
Методы				
Conjugate	public	Quaternion		Сопряженный кватернион
Reciprocal	public	Quaternion		Обратный кватернион
Normalize	public	Quaternion		Нормализация
Round	public	Quaternion	Int32	Округление до определенного количества знаков после запятой
Approximate	public	Quaternion	Single	Округление до определенной точности
Rotation Quaternion	public static	Quaternion	Quaternion, Single	Вычисление кватерниона поворота для заданных оси и угла в радианах
Rotate	public static	Quaternion	Quaternion, Quaternion,	Поворот векторного кватерниона на заданные

			Single	ось и угол в радианах
Rotate	public static	Quaternion	Quaternion, Quaternion, Quaternion, Single	Поворот векторного кватерниона на заданные ось (со смещением) и угол в радианах
Rotate Composition	public static	Quaternion	Quaternion, Quaternion, Single Quaternion, Single	Комбинирование двух поворотов
Slerp	public static	Quaternion	Quaternion, Quaternion, Single	Сферическая интерполяция (SLERP) между двумя кватернионами
Lerp	public static	Quaternion	Quaternion, Quaternion, Single	Линейная интерполяция между двумя кватернионами
LSlerp	public static	Quaternion	Quaternion, Quaternion, Single	Приближенная сферическая интерполяция (получается из линейной по специальному алгоритму; процедура является гораздо менее ресурсоемкой, чем Slerp)
ToString	public override	String		Строковое представление кватерниона
ToString	public override	String	Int32	Строковое представление с определенной точностью
Equals	public override	Boolean	Object	Проверка на равенство
GetHashCode	public override	Int32		Хэш-код объекта
Приведение типов				
Quaternion	implicit	Quaternion	Single	
Quaternion	implicit	Quaternion	Vector3	
Vector3	explicit	Vector3	Quaternion	

3.1.2. Структура Fraction

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
Empty	public static readonly	Fraction		«Пустая» дробь
Свойства				
Numerator	public	Int64		Числитель
Denominator	public	UInt32		Знаменатель
Конструкторы				
Fraction	public		Int64, UInt32	Конструктор общего вида
Методы				
Cancel	public	Fraction		Сокращение дроби

Reverse	public	Fraction		Переворачивание дроби
ReduceToDenominator	public	Fraction	UInt32	Приведение дроби к знаменателю
ToMixedNumber	public	void	out Int64, out Fraction	Получение из дроби смешанного числа
FromMixedNumber	public static	Fraction	Int64, Fraction	Создание дроби из смешанного числа
Parse	public static	Fraction	String	Создание дроби путем анализа строки
TryParse	public static	Boolean	String, out Fraction	Создание дроби путем анализа строки, не вызывающее исключений
IsEmpty	public static	Boolean	Fraction	Проверка дроби на равенство «пустой»
ToString	public override	String		Строковое представление дроби
Equals	public override	Boolean	Object	Проверка на равенство
GetHashCode	public override	Int32		Хэш-код объекта
CompareTo	public	Int32	Object	Реализация интерфейса IComparable
Приведение типов				
Fraction	implicit	Fraction	Int64	
Fraction	explicit	Fraction	Double	
Double	explicit	Double	Fraction	

3.1.3. Структура Angle

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
PI	public static readonly	Angle		Величина 180°
Eps	public static readonly	Angle		Величина 0°0'1''
Свойства				
Sign	public	Int32		Знак градусной величины
Degrees	public	UInt32		Градусы
Minutes	public	UInt32		Минуты
Seconds	public	UInt32		Секунды
Конструкторы				
Angle	public		UInt32, Int32	Конструктор общего вида
Angle	public		UInt32, UInt32, UInt32, Int32	Конструктор общего вида
Методы				
Parse	public static	Angle	String	Создание градусной величины путем анализа строки

TryParse	public static	Boolean	String, out Angle	Создание градусной величины путем анализа строки, не вызывающее исключений
Wrap	public static	Angle	Angle	Приведение угла к отрезку $[-180^\circ, 180^\circ]$
ToString	public override	String		Строковое представление градусной величины
ToString	public override	String	Boolean, Boolean	Строковое представление без вывода нулевых значений минут или секунд
Equals	public override	Boolean	Object	Проверка на равенство
GetHashCode	public override	Int32		Хэш-код объекта
CompareTo	public	Int32	Object	Реализация интерфейса IComparable
Приведение типов				
Double	explicit	Double	Angle	
Fraction	explicit	Fraction	Angle	
Angle	explicit	Angle	Double	
Angle	explicit	Angle	Fraction	

3.1.4. Статический класс *Common*

Имя	Модификаторы	Тип	Аргументы	Описание
Методы				
Round	public static	Double	Double, Int32	Округляет число до заданного числа знаков после запятой
Round	public static	Double	Double	Округляет число до целого по математическим правилам
Cbrt	public static	Double	Double	Вычисляет кубический корень из числа
Factorial	public static	Double	UInt32	Вычисляет факториал числа
DoubleFactorial	public static	Double	UInt32	Вычисляет двойной факториал числа
BinomialCoefficient	public static	Double	UInt32, UInt32	Вычисляет C_n^k
LCM	public static	UInt32	UInt32, UInt32	Вычисляет НОК двух чисел
GCD	public static	UInt32	UInt32, UInt32	Вычисляет НОД двух чисел
FractionalPart	public static	Double	Double	Возвращает дробную часть числа
FractionalPart	public static	Decimal	Decimal	Возвращает дробную часть числа
FractionalPartLength	public static	Int32	Double	Возвращает длину дробной части числа

FractionalPartLength	public static	Int32	Decimal	Возвращает длину дробной части числа
IsPrime	public static	Boolean	UInt32	Проверяет, является ли число простым
Swap	public static	void	ref Int32, ref Int32	Меняет местами значения двух переменных
Swap	public static	void	ref UInt32, ref UInt32	Меняет местами значения двух переменных

3.2. Библиотека *DCL.Phone.Xna*

3.2.1. Абстрактный класс *Shape*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
current Vertices	protected	VertexPosition Normal Texture[]		Массив вершин модели (в том числе после преобразований)
startVertices	protected	VertexPosition Normal Texture[]		Массив вершин модели до всех преобразований
triangle Indices	protected	Int16[]		Массив индексов вершин модели для отрисовки ее по треугольникам
lineIndices	protected	Int16[]		Массив индексов вершин модели для отрисовки ее контура по линиям
startCenter	protected	Vector3		Центр фигуры до всех преобразований
Свойства				
Center	public	Vector3		Центр фигуры
Texture	public	Texture2D		Текстура модели
Graphics Device	public	Graphics Device		Графическое устройство, на которое отрисовывается модель
Default Rotation	protected	Quaternion		Стандартное вращение, заданное кватернионом
Конструкторы				
Shape	protected			Конструктор общего вида
Методы				
Reset	public	void		Возвращает модель к ее первоначальному состоянию
Rotate	public	void	Quaternion, Single	Поворачивает модель вокруг некоторой оси на некоторый угол
Rotate	public	void	Quaternion, Quaternion, Single	Поворачивает модель вокруг некоторой смещенной оси на некоторый угол
Rotate Composition	public	void	Quaternion, Single, Quaternion, Single	Осуществляет комбинирование двух поворотов модели
Rotate Composition	public	void	Quaternion, Single, Quaternion, Single, Quaternion,	Осуществляет комбинирование трех поворотов модели

			Single	
RotateDefault	public	void		Осуществляет стандартный поворот модели
SetDefault Rotation	public	void	Quaternion, Single	Задаёт стандартный поворот модели
SetDefault Rotation	public	void	Quaternion, Single, Quaternion, Single	Задаёт стандартный поворот модели как комбинацию двух поворотов
SetDefault Rotation	public	void	Quaternion, Single, Quaternion, Single, Quaternion, Single	Задаёт стандартный поворот модели как комбинацию трех поворотов
Translate	public	void	Quaternion	Осуществляет перенос модели в пространстве на некоторый вектор
Scaling	public	void	Single	Осуществляет масштабирование модели относительно ее центра
Draw	public	void	BasicEffect, DrawMode	Рисует модель на графическом устройстве
DrawScene	public static	void	BasicEffect, DrawMode, params Shape[]	Осуществляет отрисовку нескольких моделей одновременно
DrawScene	public static	void	BasicEffect, DrawMode, Shape[], params Shape[]	Осуществляет отрисовку нескольких моделей одновременно

3.2.2. Класс *Ellipse* – наследник *Shape*

Имя	Модификаторы	Тип	Аргументы	Описание
Свойства				
Radius	public	Single		Базовый радиус эллипса
Axis	public	Vector3		Ось эллипса
Конструкторы				
Ellipse	public		Vector3, Single, Single, Single, Int32	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей.

Ellipse	public		Vector3, Single, Single, Single, Int32, Texture2D	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей, текстура модели.
Ellipse	public		Vector3, Single, Single, Single, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей, текстура модели и стандартное графическое устройство.

3.2.3. Класс *Circle* – наследник *Ellipse*

Имя	Модифика- торы	Тип	Аргументы	Описание
Конструкторы				
Circle	public		Vector3, Single, Int32	Конструктор общего вида. Задаются центр модели, степень детализации и радиус окружности.
Circle	public		Vector3, Single, Int32, Texture2D	Конструктор общего вида. Задаются центр модели, степень детализации, радиус окружности и текстура модели.
Circle	public		Vector3, Single, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются центр модели, степень детализации и радиус окружности, текстура модели и стандартное графическое устройство.

3.2.4. Класс *Ellipsoid* – наследник *Shape*

Имя	Модифика- торы	Тип	Аргументы	Описание
Свойства				
Radius	public	Single		Базовый радиус эллипсоида
Axis	public	Vector3		Ось эллипсоида
Конструкторы				
Ellipsoid	public		Vector3, Single, Single,	Конструктор общего вида. Задаются центр модели,

			Single, Single, Int32	базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей.
Ellipsoid	public		Vector3, Single, Single, Single, Single, Int32, Texture2D	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей, текстура модели.
Ellipsoid	public		Vector3, Single, Single, Single, Single, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей, текстура модели и стандартное графическое устройство.

3.2.5. Класс *Sphere* – наследник *Ellipse*

Имя	Модификаторы	Тип	Аргументы	Описание
Конструкторы				
Sphere	public		Vector3, Single, Int32	Конструктор общего вида. Задаются центр модели, степень детализации и радиус сферы.
Sphere	public		Vector3, Single, Int32, Texture2D	Конструктор общего вида. Задаются центр модели, степень детализации, радиус сферы и текстура модели.
Sphere	public		Vector3, Single, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются центр модели, степень детализации и радиус сферы, текстура модели и стандартное графическое устройство.

3.2.6. Класс *Dot* – наследник *Shape*

Имя	Модификаторы	Тип	Аргументы	Описание
Конструкторы				
Dot	public		Vector3, Int32	Конструктор общего вида. Задаются положение точки и степень ее

				«жирности».
Dot	public		Vector3, Int32, Texture2D	Конструктор общего вида. Задаются положение точки, степень ее «жирности» и текстура.
Dot	public		Vector3, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются положение точки, степень ее «жирности», текстура и стандартное графическое устройство.

3.2.7. Перечисление *DrawMode*

- Lines – рисуется контур модели;
- Solid – рисуется поверхность модели;
- SolidWithLines – рисуются как поверхность, так и контур модели;
- Textured – рисуется поверхность модели с натянутой на нее текстурой.

3.2.8. Класс *PivotGame* – наследник *Microsoft.Xna.Framework.Game*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
HeaderFont	protected internal	SpriteFont		Шрифт заголовка страниц приложения
ContentFont	protected internal	SpriteFont		Шрифт текста содержимого приложения
TitleFont	protected internal	SpriteFont		Шрифт заголовка приложения
Selection Changed Event Handler	public	delegate void		Делегат обработчика события SelectionChanged
Selection Changed	public	event Selection Changed EventHandler		Событие, возбуждаемое при смене страницы приложения
Свойства				
Title	public	string		Заголовок приложения
Background Color	public	Color		Цвет фона страниц приложения
Foreground Color	public	Color		Цвет текста страниц приложения
Scene Background Color	public	Color		Цвет фона области, в которую отрисовывается трехмерная сцена
Scene Background Texture	public	Texture2D		Фоновая текстура области, в которую отрисовывается

				трехмерная сцена
DrawingArea	public	Rectangle		Область, в которую отрисовывается трехмерная сцена
SplashScreen Image	public	Texture2D		Картинка-заставка на время загрузки приложения
Graphics Device Manager	protected internal	Graphics Device Manager		Менеджер графического устройства
SpriteBatch	protected internal	SpriteBatch		Менеджер спрайтов, отвечающий за отрисовку текстур и текста
BasicEffect	protected	BasicEffect		Эффект, используемый при отрисовке сцены
Projection Type	protected	Projection		Тип проекции, используемой при отрисовке сцены
CameraScale	public	Single		Степень охвата сцены камерой
Zoom	public	Single		Степень удаленности камеры от начала координат по оси Z
SceneCenter Translation	protected	Vector2		Центр сцены
SelectedIndex	public	Int32		Номер текущей страницы приложения
<i>indexer</i>	public	PivotGame Item	Int32	Индексатор, возвращающий страницу приложения по ее номеру
ItemsCount	public	Int32		Количество страниц приложения
Changing Pages	public	Boolean		Определяет, происходит ли в данный момент смена страниц
Touches	protected	Touch Collection		Массив касаний экрана пользователем в каждый момент времени
Конструкторы				
PivotGame	public			Конструктор общего вида
Методы				
AddItem	public	void	PivotGame Item	Добавляет страницу в приложение
RemoveItem	public	void	PivotGame Item	Удаляет страницу из приложения
DrawString	public	void	SpriteFont, String, Vector2, Color	Отрисовывает строку на заданной позиции
DrawSprite	public	void	Texture2D, Vector2, Color	Отрисовывает текстуру на заданной позиции

3.2.9. Класс *PivotGameItem*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
DrawingArea	public	DrawingArea		Область рисования сцены
DrawFrame	public	event EventHandler		Событие, возбуждаемое при перерисовке страницы (за исключением сцены)
DrawScene	public	event EventHandler		Событие, возбуждаемое при перерисовке сцены на страницы
Свойства				
BackgroundColor	public	Color		Цвет фона страницы
ForegroundColor	public	Color		Цвет текста страницы
Header	public	String		Заголовок страницы
Parent	public	PivotGame		Ссылка на объект класса приложения, в состав которого входит страницы
Конструкторы				
PivotGameItem	public			Конструктор общего вида

3.2.10. Класс *SelectionChangedEventArgs* – наследник *EventArgs*

Имя	Модификаторы	Тип	Аргументы	Описание
Свойства				
PreviousIndex	public	Int32		Индекс предыдущей страницы.
Конструкторы				
SelectionChangedEventArgs	public		Int32	Конструктор общего вида. Задается индекс предыдущей страницы.

3.2.11. Структура *DrawingArea*

Имя	Модификаторы	Тип	Аргументы	Описание
Свойства				
X	public	Int32		Координата X верхнего левого угла области отрисовки сцены
Y	public	Int32		Координата Y верхнего

				левого угла области отрисовки сцены
Width	public	Int32		Ширина области отрисовки сцены
Height	public	Int32		Высота области отрисовки сцены
Background Color	public	Color		Цвет фона области отрисовки сцены
Background Texture	public	Texture2D		Фоновая текстура области отрисовки сцены

3.2.12. Перечисление *ProjectionType*

- Perspective – перспективная проекция;
- Orthographic – ортографическая проекция.

4. Входные и выходные данные

4.1. Организация входной информации

Для задания объекта, представляющего кватернион, на вход программе требуется четыре вещественных числа одинарной точности (типа *System.Single*). Возможно создание объекта-кватерниона через конструкторы приведения типов *System.Int32* (для создания чисто скалярного кватерниона) и *Microsoft.Xna.Framework.Vector3* (для создания чисто векторного кватерниона).

Для задания объекта, представляющего обыкновенную дробь, на вход программе требуется два целочисленных значения (для числителя и знаменателя).

Для задания объекта, представляющего градусную величину, на вход программе требуется три целочисленных значения (для представления градусов, минут и секунд).

Для создания объекта, представляющего трехмерную модель, на входе необходимо указать ее местоположение в пространстве (объект типа *Microsoft.Xna.Framework.Vector3*), степень детализации, в случае эллипсоида, сферы, эллипса или окружности – радиус. Опционально при создании объекта в конструкторе можно указать ссылку текстуру для модели. Текстура должна представлять собой объект типа *Microsoft.Xna.Framework.Texture2D*.

Для осуществления операции поворота модели методу *Rotate* класса *DCL.Phone.Xna.Shape* на входе требуется информация об оси поворота (объект типа *DCL.Maths.Quaternion*) и об угле поворота в радианах (вещественное число типа *System.Single*).

4.2. Организация выходной информации

Выходной информацией является изображение построенной и преобразованной (например, поворотом через кватернионы) трехмерной модели на экране. Визуализация осуществляется посредством вызова метода *Draw* в классах, производных от класса *DCL.Phone.Xna.Shape*.

5. Сообщения

Возникающие в ходе работы библиотеки исключительные ситуации не обрабатываются непосредственно библиотеками. Ответственность за контроль и обработку исключительных ситуаций несет программист, использующий библиотеки в своем

программном продукте. Во время выполнения в библиотеке могут возникать следующие исключительные ситуации:

- В случае обнаружения некорректных входных данных (отрицательных значений радиуса сферы или параметра детализации) в конструкторах классов, представляющих трехмерные модели, возбуждается исключение *ArgumentOutOfRangeException*;
- В случае попытки создания объекта-дроби со знаменателем, равном 0, возбуждается исключение *DivideByZeroException*.