|  |  |
| --- | --- |
|  |  |
|  |

ИСТОРИЯ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Версия** | **Описание изменений** | **Автор** | **Дата** | **Утверждено**   |  |  | | --- | --- | | **Имя** | **Дата** | | |
| <1.0> | Initial version | Alexander\_Kuznetsov1 | <20-Jan-2012> |  |  |
|  |  |  |  |  |  |

СВЯЗАННЫЕ ДОКУМЕНТЫ

|  |  |
| --- | --- |
| **Номер** | **Название документа** |
|  |  |
|  |  |

# Задания

# Полиморфизм

## Задание 1

Напишите заготовку для векторного графического редактора. Полная версия редактора должна позволять создавать и выводить на экран такие фигуры как: Линия, Окружность, Прямоугольник, Круг, Кольцо. Заготовка, для упрощения, должна представлять собой консольное приложение с функционалом:

1. Создать фигуру выбранного типа по произвольным координатам.
2. Фигуры должны создаваться в общей коллекции (массиве)
3. Вывести фигуры на экран (для каждой фигуры вывести на консоль её тип и значения параметров реализовать в методе Draw) с использованием перегрузки метода Draw

## Задание 2

За основу для задания взять классы и интерфейсы из демонстрации, файл «07\_Interfaces.cs». Требуется реализовать интерфейс «ISeries» таким образом, чтобы получить геометрическую прогрессию. Продемонстрировать его работу с помощью метода PrintSeries.

Два свойства в интерфейсе

Дабл – первый элемент

Бул – следующий

// Упаковка структуру записать как интерфейс

## Задание 3

За основу для задания взять классы и интерфейсы из демонстрации, файл «07\_Interfaces.cs».

Требуется:

* Создать интерфейс «IIndexable», позволяющий обраться к элементу последовательности по индексу. (см. индексаторы)
* Классы «ArithmeticalProgression» и «List» должны реализовывать интерфейс «IIndexable» одновременно с «ISeries».
* Создать метод, демонстрирующий работу с объектами «IIndexable»

## Задание 4\*

Создайте иерархию классов и пропишите ключевые методы для компьютерной игры (без реализации функционала). Суть игры:

1. Игрок может передвигаться по прямоугольному полю размером Width на Height.
2. На поле располагаются бонусы (яблоко, вишня и т.д.), которые игрок может подобрать для поднятия каких-либо характеристик.
3. За игроком охотятся монстры (волки, медведи, когти смерти и т.д.), которые могут передвигаться по карте по какому-либо алгоритму.
4. На поле располагаются препятствия разных типов (камни, деревья и т.д.), которые игрок и монстры должны обходить.

Цель игры – собрать все бонусы и не быть «съеденным» монстрами.

Примеры Солид

1. Единственная ответственность модулей
2. Открытость для расширения, закрыты для модификации.

3.

4. Принцип подстановки наследников

Приватный \_х

Х только на чтение, возвращает \_х.

Для у тоже

Конструктор принимает х и у.

А в квадрате конструктор принимает только это число

И базовый вызывает два раза и принимает это число :base (x,x)

1. Принцип инверсии зависимостей

Не завязывать свой класс на отдельную реализацию. Стараться своему класс у передать также и вывод в консоль и вывод в файл, например.