

Задание 1.

Создать абстрактный класс Polygon, содержащий следующие элементы: закрытые поля – название фигуры, массив сторон; конструктор с параметрами, метод ToString(), абстрактный метод вычисления площади многоугольника. Реализовать в этом классе интерфейс IComparable для сравнения многоугольников по площади.

Создать класс Triangle, производный от класса Polygon, содержащий свойства для чтения сторон Side1, Side2, Side3, реализовать в нем метод вычисления площади и добавить метод для вычисления периметра.

Создать класс Rectangle, производный от класса Polygon, содержащий свойства для чтения сторон Width и Height, реализовать в нем метод вычисления площади.

В программе нужно формировать массив многоугольников (не меньше четырех), состоящий из объектов Triangle и Rectangle (не вводить с клавиатуры), выводить информацию в табличном виде.

Массив нужно отсортировать по убыванию площадей.

Фигура	Стороны	Площадь	Периметр
треугольник	5,3,4	6	12
прямоугольник	4,2	8	-
...			

Определить количество фигур, площадь которых меньше заданной.

Задание 2.

Заменить все большие буквы в исходной строке маленькими. После каждого слова вставить символ \$. Сформировать массив строк String из предложений текста, в которых все слова состоят не больше чем из пяти символов. Разделителями слов считать пробел, запятую, точку с запятой. Разделителями предложений считать точку, восклицательный и вопросительный знаки, символ конца строки.

Задание 3 (необязательное, было давно на тренинге ЕРАМ).

У предпринимателя имеется паром, который перевозит грузы и людей. Поля для класса Паром: грузоподъемность и общий массив для грузов и людей.

Виды транспортировки грузов:

– прямоугольные контейнеры для однородного твердого материала (масса на основе плотности материала, параметров контейнера и массы самого контейнера);

– платформы для любого груза (масса на основе массы груза; платформа является частью паромы, т.е. ее масса принимается равной 0);

– цилиндрические цистерны для жидкости (масса на основе плотности жидкости, параметров цистерны и массы самой цистерны).

Спроектировать объектную модель.

В методе *main* создать необходимые объекты (ничего не вводить, использовать вызовы конструкторов с константными значениями).

Вывести содержимое общего массива.

Отсортировать содержимое массива по людям и видам транспортировки грузам (сначала люди, затем грузы в контейнерах, на платформах, в цистернах).

Вывести содержимое общего массива.

Определить, может ли перевезти паром заданные грузы и людей.

Ограничение: не делать общей иерархии классов для людей и грузов (это не логично). Подсказка: данные классы должны реализовывать одинаковый интерфейс, через которые их можно объединить в массив.

Замечания

– Дополнение к code conventions.

Идентификаторы булевских методов имеют вид `IsXxx()`, где `Xxx` – прилагательное или причастие.

Пример.

```
public boolean IsPassed()
{
    return true;
}
```

– Ради удобства решения задачи человека хочется приравнять к грузу. Однако это неестественно с точки зрения житейской логики. Поэтому ООП-гуру не советуют так делать. Всегда нужно думать о развитии проекта. Если возникнет у заказчика желание получить новый функционал (что есть обычная практика при его платежеспособности!), то раньше или позже НЕЕСТЕСТВЕННАЯ ИЕРАРХИЯ с большой вероятностью вылезет боком. Таким образом, ограничение о разной иерархии классов для людей и грузов является разумным.

Кстати, почитайте мастера на http://skipy.ru/architecture/module_design.html.