



Практическая работа 12.

Обобщенные типы

Задание 1

Требуется создать класс `MyStack<T>`, который будет реализовывать структуру коллекции данных с использованием обобщенных типов. Стек должен поддерживать операции **добавления в конец** элемента (Push), **извлечения и возвращение** последнего добавленного элемента (Pop) и **проверки наличия в массиве элементов** (IsEmpty). Элементы должны храниться в массиве.

Задание 2

Создайте классы Person, Food, Transport и **переопределите в них ToString()**, возвращая информацию об объектах.

Напишите в Main обобщенный метод `GenerateReport<T>(T[] array)`, который принимает **коллекцию объектов типа T** и выводит список объектов в консоль. Для вывода объекта, необходимо использовать метод `ToString()`.

Для проверки передайте в метод массивы string, int, Person, Food, Transport.

Задание 3

Требуется разработать систему для описания характеристик персонажей. Для этого создайте класс `Stat<T>`, который описывает общую характеристику персонажа, такую как здоровье, сила атаки, ловкость и т.п.



Класс должен хранить значение в **защищенном универсальном поле** `_value`.

Класс должен иметь **публичное виртуальное универсальное свойство** `Value`
`{get;set;}`

Однако, разные характеристики могут измеряться различными способами:

1. Дробные числа, выражаются в процентах от 0 до 100 (например, шанс уклонения);
2. Целые числа, ограничиваются диапазоном между 0 и максимальным значением (например, количество здоровья).

Создайте соответствующие производные от `Stat<T>` классы с [явным определением обобщенных типов](#) `float` и `int`. Создайте производный от `Stat<T>` класс `ItemStat<T>`, который [ограничивает обобщенный тип](#) классом `Item`.

Необходимо создать базовый класс `Item`, представляющий любые игровые предметы, и два производных класса:

- **Armor**: броня, обладающая характеристиками защиты.
- **Weapon**: оружие, которое обладает характеристикой урона.

Создайте класс `Hero`, который будет представлять героя игры. Герой имеет следующие характеристики:



- **Health**: здоровье, ограниченное значением от 0 до 200.
- **DodgeChance**: шанс уклонения, выраженный в процентах.
- **Armor**: броня, предоставляющая защиту.
- **Weapon**: оружие, определяющее атаку.

Все эти характеристики должны храниться в соответствующих экземплярах классов **Stat<T>** или его производных реализациях.

Контрольные вопросы:

1. [Что такое универсальный параметр?](#)
2. [Как указать универсальный параметр? Как указать несколько универсальных параметров?](#)
3. [Что такое универсальный метод?](#)
4. [Как ограничить универсальный параметр у метода?](#)
5. [Как ограничить универсальный параметр у класса?](#)
6. [Чем можно ограничить универсальные параметры?](#)
7. [Как ограничить несколько универсальных параметров?](#)
8. [Как наследовать универсальный параметр?](#)
9. [Как в производном классе явно объявить тип универсального параметра?](#)