**Лабораторная работа №10. Часть 2**

*«Классы. Конструкторы, методы и свойства»*

**Рекомендации по программированию**

* Поля предпочтительнее делать private.
* Интерфейс должен быть полным, т.е. предоставлять возможность выполнять любые разумные действия с классом. И одновременно минимально необходимым, т.е. без дублирования и пересечения возможностей методов.
* Каждый метод класса должен решать только одну задачу. Не стоит объединять два коротких независимых фрагмента кода в один метод.
* Если действия встречаются хотя бы дважды, стоит оформить их в отдельные функции.
* Все заданные методы, поля и экземпляры классов должны иметь осмысленные имена.

**Требования к лабораторной**

* Каждый разрабатываемый класс должен, как правило, содержать следующие элементы: поля с различными спецификаторами, конструкторы с параметрами и без параметров, методы, свойства.
* Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса.
* При возникновении ошибок должны выбрасываться исключения
* Каждый студент должен реализовать свой класс со своими полями и методами. Любое совпадение не случайно и карается по закону джунглей.
* Проделанную работу необходимо показать на паре.

**Постановка задачи**

1. Создать класс, который является некоторой произвольной расой в игре. Объявить не менее десяти полей-характеристик для данного класса, используя различные спецификаторы доступа. Реализовать механизмы ввода и вывода данных для одного персонажа этого класса.

*Пример: Класс Псайдак с полями: имя, пол, вес, рост, цвет, сила, физмастерство, уклонение, лечение, арсенал, боевая способность (разгромный удар), не боевая способность (телекинез) и другие.*

1. Для предыдущего задания реализовать все возможные типы конструкторов (не менее трех). Реализовать свойства, позволяющие осуществить проверку на допустимые значения для данных класса. Создать массив персонажей данного класса. Каждый персонаж должен быть уникальным. Вывести этот массив на консоль.
2. Для предыдущего задания реализовать не менее шести **полезных** методов для работы с заданным классом. Также реализовать методы сортировки по каждому имеющемуся полю, вывода максимального и минимального значения поля (где таковое возможно).

*Например, поиск оптимального удара среди ударов ногой, рукой, головой (максимальный урон от удара, который придется соперника) у каждого персонажа этой расы. Методы для расчета ударов ногой, рукой и головой должны быть отдельные и свои.*

В отчете после формулировки заданий подписывать созданные в коде поля и методы. **Пример**:

1. Создать класс, который является некоторой произвольной расой в игре. Объявить не менее десяти полей-характеристик для данного класса, используя различные спецификаторы доступа. Реализовать механизмы ввода и вывода данных для одного персонажа этого класса.

*Класс Psyduck с полями: Name (имя), Gender (пол), Weight (вес), Height (рост), Color (цвет), Power (сила), PhysicalMastery (физмастерство), Evasion (уклонение), Healing (исцеление) и т.д.*

1. Для предыдущего задания реализовать не менее шести **полезных** методов для работы с заданным классом. Также реализовать методы сортировки по каждому имеющемуся полю, вывода максимального и минимального значения поля (где таковое возможно).

*Методы: OpimalHit (оптимальный удар), Kick (удар ногой), HeadButt (удар головой), Punch (удар рукой) и т.д.*

Псы: А еще можете пописать названия переменных, классов и методов на русском. Это забавно)