1) Перенос методов вычисления возраста в специальный класс, чтобы не загромождать менеджеры логикой, которая непосредственно к ним не относится.

2) Необходимо использование [PortalMvcExceptionFilter] для отлавливания исключений в контроллере

3) добавить Id в модель

Вместо public async Task<ActionResult> OwnerDetails(OwnerDetailsViewModel post, int ownerId)

использовать

public async Task<ActionResult> OwnerDetails(OwnerDetailsViewModel post),

а поле Id создать в OwnerDetailsViewModel

4) Методы должны возвращать значения. Вместо

public async Task SetHorseOwnerParameters(HorseOwnerDto owner, HorseOwnerStatsFilter filter)

Использовать

public async Task<HorseOwnerDto> GetHorseOwnerParameters(HorseOwnerDto owner, HorseOwnerStatsFilter filter)

5) Использовать более стандартные и понятные имена, например, GetHorse вместо GetHorseDto и GetFullHorseInfo вместо GetOrderedHorseDto

6) Переменные следует объявлять тогда, когда их использование необходимо или способствует повышению читабельности кода. В остальных случаях предпочтительнее обходиться без лишних объявлений, например, вместо

var filter = prizeViewModelDto.Filter;

var prizeList = await \_prizeMgr.GetPrizesList(filter);

prizeViewModelDto.Prizelist = prizeList;

компактнее и удобнее запись

prizeViewModelDto.Prizelist = await \_prizeMgr.GetPrizesList(prizeViewModelDto.Filter);

7) LINQ - выражения следует записывать следующим образом: в каждой строке должна быть одна операция. Вместо

outputInfo.Owner.TestedHorsesCount = owner.OwnerResults.Select(os => os.HorseId).Distinct().Count();

Читабельнее

outputInfo.Owner.TestedHorsesCount = owner

.OwnerResults

.Select(os => os.HorseId)

.Distinct()

.Count();

8) Следует использовать CodeContracts для проверки входящих значений, например

public async Task<OwnerDetailsModel> GetOwnerDetails(HorseOwnerStatsFilter filter, int ownerId)

{

Contract.Requires<ArgumentException>(ownerId > 0);

var owner = await GetOwnerById(ownerId);

Contract.Requires<ArgumentNullException>(owner != null);

9) Для получения int? следует использовать метод GetValueOrDefault

10) Следует следить за выполнением принципа DRY (Don`t Repeat Yourself). Например, часто Get и Post методы содержат повторяющийся код, который имеет смысл вынести в отдельный метод