

Конструирование программного обеспечения

Домашнее задание №1. Закрепление принципов SOLID, внедрения зависимостей средствами DI-контейнера.

Формат сдачи ДЗ определяется семинаристом. По умолчанию, ДЗ выполняется на языке программирования, который является основным на семинарских занятиях, но по согласованию с семинаристом можно выполнить на другом языке программирования.



Предисловие.

Московский зоопарк обратился Высшую школу экономики с просьбой разработать ERP систему предприятия, которая обеспечит автоматизацию, интеграцию и интеллектуальность, необходимые для эффективного выполнения всех повседневных задач.

Со слов заказчика записано: «Мы хотим сделать наших животных чуть более счастливее! Счастливое животное – в первую очередь сытое и здоровое животное! Нам важно, что бы они получали еду вовремя, а также, согласно внутреннему регламенту, проходили медосмотр при приеме в наш зоопарк».

Перед вами поставлена задача разработать доменную модель классов с соблюдением принципов SOLID, а также с применением DI-контейнера

В ходе дальнейшего общения с заказчиком были выявлены детали, которые помогут вам спроектировать доменную модель классов. ERP-система в будущем будет развиваться, и в конечном счете производить учет не только животных, но и людей, а также инвентаризировать вещи состоящие на балансе предприятия. В ходе беседы мы выяснили, что инвентаризации подлежат не только предметы зоопарка, но и животные (но не сотрудники). У всех живых существ нужно хранить информацию о том какое количество килограммов еды они потребляют в сутки, а у травоядных еще и информацию об уровне их доброты, так как если доброта животного выше 5 баллов (из 10 возможных), то с данным животным смогут интерактивно взаимодействовать посетители. Архитектор проекта не сильно помог вам с доменной моделью классов, но кое-какие идеи по структуре типов он озвучил на кик-офф встрече по проекту. Вы можете воспользоваться его идеями по своему усмотрению. Он предлагает создать:

- `interface IAlive` – для определения принадлежности наших типов к категории «живых». Интерфейс будет наделять реализуемые его типы свойством `int Food` – для учета потребляемого количества еды в кг в сутки.
- `interface IInventory` – для определения принадлежности наших типов к категории «инвентаризационная вещь». Интерфейс будет наделять реализуемые его типы свойством `int Number`

В следствии того, что в будущем, каждое животное будет обладать своим собственным набором уникальных характеристик архитектор настоял на реализации следующих классов:

```
class Animal, class Herbo, class Predator, class Monkey, class Rabbit, class Tiger, class Wolf
```

Класс описывающий зоопарк, будет иметь ассоциацию с классом ветеринарная клиника, которая проверяет новых животных и принимает решение о включение в коллекцию животных зоопарка на основании здоровья животного.

Иерархия вещей пока не проработана с заказчиком, но для апробации доменной модели архитектор предлагает создать также несколько начальных классов для дальнейшей иерархии:

```
class Thing, class Table, class Computer
```

ТРЕБУЕТСЯ

1. Создать консольное приложение, которое позволит вести учет животных Московского зоопарка. Принимать их на баланс, выводить отчет по количеству животных, потребляемому ими килограммов еды в день, формировать списки тех животных, которые могут быть помещены в контактный зоопарк. Выводить наименование и инвентаризационные номера вещей, животных, стоящих на балансе зоопарка.
2. Написать readme-файл с пояснениями идей вашего решения (можно кратко, но что б из него была понятна логика принимаемых вами решений). Также, добавить инструкция по запуску.

Функциональные требования к приложению

1. Обеспечить возможность добавления новых животных в зоопарк.
2. Перед приемкой нового животного необходимо проверить его состояние здоровья (здоров / не здоров). Здоровье проверяет ветеринарная клиника, которая в ходе осмотра принимает одно из двух решений: принять животное в зоопарк или отказаться от данного животного.
3. Необходимо иметь возможность информировать о том какое количество килограмм еды нужно всем животным.
4. Необходимо иметь возможность предоставить информацию о тех животных, которые могут быть отправлены в контактный зоопарк.

Критерии оценки

1. Применение принципов ООП и SOLID для структурирования кода - 4 балла
2. Наличие readme с пояснениями, где вы применили тот или иной принцип SOLID - 1 балл
3. Применение DI-контейнера - 1.5 балла
4. Покрытие кода юнит- тестами (не менее 60% покрытия) - 2 балла
4. Понятный ввод/вывод информации на консоль (0.5)
5. Code style - 1 балл

Штрафы

1. до – 2 баллов за наличие ошибки во время выполнения кода;
2. до – 5 баллов, если программа не собирается;
3. предусмотрено снижение оценки пропорционально объему невыполненных функциональных требований.