Міністерство освіти та науки України

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

**ЗВІТ**

про виконання

лабораторної роботи №1

на тему:

**«Шаблоны проектирования. Порождающие шаблоны»**

## з дисципліни «Веб-орієнтована розробка ПЗ»

**Варіант №7**

**Виконав:** студент групи ІС-62

Ковинєв Кирило Олексійович

**Перевірила:** доцент каф. АСОІУ,

Ліщук Катерина Ігорівна

Київ 2018

**Постановка задачи к компьютерному практикуму № 1**

При выполнении компьютерного практикума следует реализовать следующие действия:

1. Реализовать задачу согласно вариантов, приведенных ниже (можно предложить реализацию собственной задачи)
2. Кратко описать архитектуру проекта
3. Привести UML-диаграмму классов
4. Описать используемые порождающие паттерны. Для каждого паттерна  
   нужно указать составные классы и их назначение

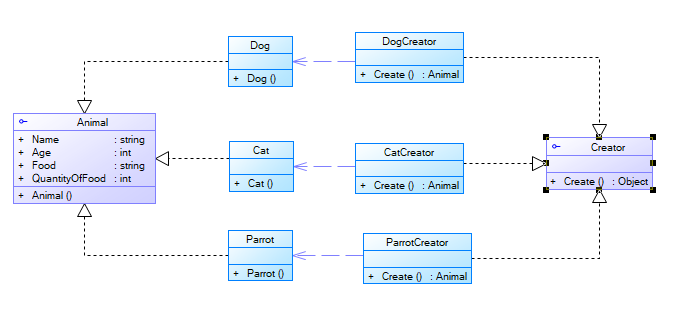
**Тема задачи:**

Реализовать задачу «Питание животных». Должна быть реализована возможность создания различных животных, с указанием, например, возраста, кол-во съедаемой еды

**Архитектура проекта**

С помощью консоли мы можем внести информацию о питании для трех различных животных. А именно: имя питомца, его возраст, количество еды, потребляемой животным, и само описание еды. Консоль предлагает меню, в котором вы выбираете нужного питомца (либо выход) и заносите информацию про него. Исходя из приведенной информации, программа создает объект с заполненными полями.

**UML-диаграмма**



**Используемый порождающий паттерн**

Фабричный метод (Factory Method) - это паттерн, который определяет интерфейс для создания объектов некоторого класса, но непосредственное решение о том, объект какого класса создавать происходит в подклассах. То есть паттерн предполагает, что базовый класс делегирует создание объектов классам-наследникам.

### Когда надо применять паттерн:

* Когда заранее неизвестно, объекты каких типов необходимо создавать
* Когда система должна быть независимой от процесса создания новых объектов и расширяемой: в нее можно легко вводить новые классы, объекты которых система должна создавать.
* Когда создание новых объектов необходимо делегировать из базового класса классам наследникам