Міністерство освіти і науки України

Національний технічний університет України „КПІ”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки

інформації та управління

**ЗВІТ**

з комп’ютерного практикуму № 1

на тему :

„ № 1. Шаблоны проектирования.

Порождающие шаблоны

”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виконав студент** |  | *ІП-63 Зозуля А.В.* |  |  |
|  |  | (№ групи, прізвище, ім’я, по батькові ) |  |  |
|  |  |  |  |  |
| **Прийняв** |  |  |  |  |
|  |  | (посада, прізвище, ім’я, по батькові ) |  |  |

Київ 2018

**ЗМІСТ**

[**1**](#_3dy6vkm) **Мета роботи 3**

[**2**](#_1t3h5sf) **Постановка задачі 4**

[**3**](#_4d34og8) **UML ДІАГРАМА КЛАСУ 5**

[**4**](#_2s8eyo1) **Висновок 6**

[**5**](#_17dp8vu) **Тескт програмного коду 7**

# 1. Мета роботи

*Мета роботи* – ознакомиться с основными шаблонами проектирования, научиться применять их при проектировании и разработке ПО.

# 

# 

# 

# 2. Постановка задачі

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

# 3. UML ДІАГРАМА КЛАСУ

# 

# 4. Висновок

В ході виконання лабораторної роботи було здійснено розробку програмного забезпечення, використовуючи породжуючий паттерн Abstract Factory.

# 5. Тескт програмного коду

# using System;

# 

# namespace Lab1

# {

# internal class Program

# {

# private static void Main(string[] args)

# {

# var animalsFeedingFactory = new AnimalsFeeding();

# Animal cat = animalsFeedingFactory.AddCat(10, 2, "Meow", "Andrii");

# 

# Animal dog = animalsFeedingFactory.AddDog(15, 3, "GoodBoy", "Andrii");

# 

# Console.WriteLine(animalsFeedingFactory.Feed(5, cat));

# Console.WriteLine(animalsFeedingFactory.Feed(15, dog));

# }

# }

# }

# 

# namespace Lab1

# {

# public class AnimalsFeeding

# {

# public Dog AddDog(int countOfFood, int age, string alias, string nameOfOwner)

# {

# var dog = new Dog(countOfFood, age, alias, nameOfOwner);

# return dog;

# }

# 

# public Cat AddCat(int countOfFood, int age, string alias, string nameOfOwner)

# {

# var cat = new Cat(countOfFood, age, alias, nameOfOwner);

# return cat;

# }

# 

# public string Feed(int countOfFood, Animal animal)

# {

# return animal.Feed(countOfFood);

# }

# }

# }

# 

# namespace Lab1

# {

# public abstract class Animal

# {

# 

# protected int Age;

# protected string Alias;

# protected int CountOfFood;

# protected string OwnerName;

# 

# protected Animal(int countOfFood, int age, string alias, string ownerName)

# {

# CountOfFood = countOfFood;

# Age = age;

# Alias = alias;

# OwnerName = ownerName;

# }

# 

# public abstract string Feed(int count);

# }

# }

# 

# using System;

# 

# namespace Lab1

# {

# public class Dog : Animal

# {

# public Dog(int countOfFood,

# int age,

# string alias,

# string nameOfOwner)

# : base(countOfFood,

# age,

# alias,

# nameOfOwner)

# {

# }

# 

# public override string Feed(int count)

# {

# if (count == CountOfFood)

# {

# return "The pet " +

# Alias +

# " is no longer hungry and loves his master " +

# OwnerName +

# "<3";

# }

# else

# {

# if (count < CountOfFood)

# return "The pet " +

# Alias +

# " is still hungry and asks his master " +

# OwnerName +

# "to feed him more :(";

# else

# return "The pet " +

# Alias +

# " is no longer hungry, but asks his master " +

# OwnerName +

# "to feed him less :)";

# }

# }

# }

# }

# using System;

# 

# namespace Lab1

# {

# public class Cat : Animal

# {

# public Cat(int countOfFood,

# int age,

# string alias,

# string nameOfOwner)

# : base(countOfFood,

# age,

# alias,

# nameOfOwner)

# {

# }

# 

# public override string Feed(int count)

# {

# if (count == CountOfFood)

# {

# return "The pet " +

# Alias +

# " is no longer hungry and loves his master " +

# OwnerName +

# " <3\n";

# }

# else

# {

# if (count < CountOfFood)

# return "The pet " +

# Alias +

# " is still hungry and asks his master " +

# OwnerName +

# " to feed him more :(\n";

# else

# return "The pet " +

# Alias +

# " is no longer hungry, but asks his master " +

# OwnerName +

# " to feed him less :)\n";

# }

# }

# }

# }

# 