

Лабораторна робота №1

Тема: Принципи програмування. DRY, KISS, SOLID, YAGNI та ін.

Мета роботи: навчитися дотримуватися принципів програмування та обґрунтовувати їх.

Завдання 1 (Варіант 2): Виконати завдання з дотриманням відомих Вам принципів програмування.

1. Створіть систему класів для обліку зоопарку. Ви можете створювати класи для різних видів і підвидів тварин; для вольєрів різних розмірів і типів; корму для тварин; працівників зоопарку.
2. Створіть класи інвентаризації, для виведення на екран інформації про наявних тварин, кількості співробітників тощо.

При виконанні завдання я використовував мову програмування Kotlin.
При написанні коду я дотримувався принципів KISS, YAGNI, SOLID.

Лістинг програми:

```
open class Animal(val name: String, val species: String)

class Lion(name: String) : Animal(name, "Lion")
class Monkey(name: String) : Animal(name, "Monkey")
class Penguin(name: String) : Animal(name, "Penguin")

class Enclosure(val id: Int, val size: String, val animalType: String) {
    fun addAnimal(animal: Animal) {
        if (animal.species == animalType) {
            println("${animal.name} успішно додано до вольєру $id")
        } else {
            println("${animal.species} не підходить для вольєру $id")
        }
    }
}

class Food(val name: String, val quantity: Int, val animalType: String) {
    fun displayInfo() {
        println("Корм для $animalType: $name, Кількість: $quantity")
    }
}
```

					ДУ «Житомирська політехніка».24.121.17.000 – Лр. 1			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Магаз А. О.			Лабораторна робота №1		Лім.	Арк.
Перевір.								1
Керівник		Фант М. О.					ФІКТ, Гр. ІПЗ-22-1[1]	
Н. контр.								
Зав. каф.								

```

class ZooKeeper(val name: String, val age: Int, val specialty: String) {
    fun displayInfo() {
        println("Ім'я: $name, Вік: $age, Спеціалізація: $specialty")
    }
}

class Inventory(private val animals: List<Animal>, private val enclosures:
List<Enclosure>, private val foods: List<Food>, private val zooKeepers:
List<ZooKeeper>) {
    fun displayAnimalInfo() {
        println("=== Інформація про тварин ===")
        animals.forEach { println("${it.name} (${it.species})") }
    }

    fun displayZooKeeperInfo() {
        println("=== Інформація про працівників ===")
        zooKeepers.forEach { it.displayInfo() }
    }

    fun displayFoodInfo() {
        println("=== Інформація про корм ===")
        foods.forEach { it.displayInfo() }
    }

    fun displayEnclosureInfo() {
        println("=== Інформація про вольєри ===")
        enclosures.forEach { println("ID: ${it.id}, Розмір: ${it.size}, Тип тва-
рин: ${it.animalType}") }
    }
}

fun main() {
    val lion1 = Lion("Алекс")
    val monkey1 = Monkey("Геннадій")
    val monkey2 = Monkey("Святослав")
    val penguin1 = Penguin("Ковальський")
    val penguin2 = Penguin("Ріко")
    val penguin3 = Penguin("Рядовий")
    val penguin4 = Penguin("Шкіпер")

    val enclosure1 = Enclosure(1, "Великий", "Lion")
    val enclosure2 = Enclosure(2, "Середній", "Monkey")
    val enclosure3 = Enclosure(3, "Маленький", "Penguin")

    val food1 = Food("М'ясо", 10, "Lion")
    val food2 = Food("Банани", 20, "Monkey")
    val food3 = Food("Риба", 20, "Penguin")

    val zooKeeper1 = ZooKeeper("Анна", 30, "Леви")
    val zooKeeper2 = ZooKeeper("Петро", 25, "Мавпи")
    val zooKeeper3 = ZooKeeper("Анастасія", 20, "Пінгвіни")

    val inventory = Inventory(
        listOf(lion1, monkey1, monkey2, penguin1, penguin2, penguin3, penguin4),
        listOf(enclosure1, enclosure2, enclosure3),
        listOf(food1, food2, food3),
        listOf(zooKeeper1, zooKeeper2, zooKeeper3)
    )
}

```

		Магаз А. О.			ДУ «Житомирська політехніка».24.121.17.000 - Лр.1	Арк.
		Фант М. О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

inventory.displayAnimalInfo()

inventory.displayZooKeeperInfo()

inventory.displayFoodInfo()

inventory.displayEnclosureInfo()

enclosure1.addAnimal(lion1)
enclosure2.addAnimal(monkey1)
enclosure2.addAnimal(monkey2)
enclosure3.addAnimal(penguin1)
enclosure3.addAnimal(penguin2)
enclosure3.addAnimal(penguin3)
enclosure3.addAnimal(penguin4)
}

```

Завдання 2: Написати код для тестування отриманої функціональності.

1. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
2. Достатньо буде просто вивести певну інформацію, щоб показати, що класи комунікують певним чином між собою.

Результат виконання:

```

=== Інформація про тварин ===
Алекс (Lion)
Геннадій (Monkey)
Святослав (Monkey)
Ковальський (Penguin)
Ріко (Penguin)
Рядовий (Penguin)
Шкіпер (Penguin)
=== Інформація про працівників ===
Ім'я: Анна, Вік: 30, Спеціалізація: Леви
Ім'я: Петро, Вік: 25, Спеціалізація: Мавпи
Ім'я: Анастасія, Вік: 20, Спеціалізація: Пінгвіни
=== Інформація про корм ===
Корм для Lion: М'ясо, Кількість: 10
Корм для Monkey: Банани, Кількість: 20
Корм для Penguin: Риба, Кількість: 20
=== Інформація про вольєри ===
ID: 1, Розмір: Великий, Тип тварин: Lion
ID: 2, Розмір: Середній, Тип тварин: Monkey
ID: 3, Розмір: Маленький, Тип тварин: Penguin
Алекс успішно додано до вольєру 1
Геннадій успішно додано до вольєру 2
Святослав успішно додано до вольєру 2
Ковальський успішно додано до вольєру 3
Ріко успішно додано до вольєру 3
Рядовий успішно додано до вольєру 3
Шкіпер успішно додано до вольєру 3

```

		Магаз А. О.			ДУ «Житомирська політехніка».24.121.17.000 - Лр.1	Арк.
		Фант М. О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 3: Опишіть особливості дотримання принципів програмування в Вашому коді

1. Додайте файл README.md в кореневу директорію цієї лабораторної роботи. В файлі README.md опишіть дотримання окремо кожного принципу програмування, який Вам відомо, і який можна продемонструвати Вашим кодом.
2. Опис можна залишати українською або (бажано) англійською мовами.
3. Ваш опис повинен містити посилання на відповідні файли і рядки коду.
4. Як залишати посилання на свої рядки коду можна глянути [тутечки](#) (для посилання на директорію) або [тут](#) (для посилання на окремі рядки).
5. Синтаксис .md файлів документації можна знайти [туть](#) або [туть](#).
6. Для отримання максимальної оцінки Ви повинні продемонструвати мінімум 7 принципів. SOLID принципи рахуються окремо. Повний список принципів, які було розглянуто на лекції:
 - a. DRY,
 - b. KISS,
 - c. SOLID (5 окремих принципів)
 - d. YAGNI
 - e. Composition Over Inheritance
 - f. Program to Interfaces not Implementations
 - g. Fail Fast

		Магаз А. О.			ДУ «Житомирська політехніка».24.121.17.000 - Лр.1	Арк.
		Фант М. О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:

KPZ / Lab01 / README.md

ArtemMahaziPZ Update README.md cd3d046 · 1 minute ago History

Preview Code Blame 31 lines (24 loc) · 3.75 KB Code 55% faster with GitHub Copilot Raw

Принцип YAGNI полягає у тому, щоб не додавати нічого зайвого. Ось приклад використання:

```
KPZ/Lab01/src/Main.kt
Lines 21 to 25 in 21f22f9

21 class Food(val name: String, val quantity: Int, val animalType: String) {
22     fun displayInfo() {
23         println("Корм для $animalType: $name, Кількість: $quantity")
24     }
25 }
```

Тут ми маємо лише найнеобхідніше. Тобто, лише метод для виведення потрібної інформації.

KISS - це принцип, який полягає у тому, що код повинен виглядати просто та зрозуміло. Приклад використання:

```
KPZ/Lab01/src/Main.kt
Lines 56 to 75 in 21f22f9

56 val lion1 = Lion("Алекс")
57 val monkey1 = Monkey("Геннадій")
58 val monkey2 = Monkey("Святослав")
59 val penguin1 = Penguin("Ковальський")
60 val penguin2 = Penguin("Ріко")
61 val penguin3 = Penguin("Рядовий")
62 val penguin4 = Penguin("Шкінеп")
63
64 val enclosure1 = Enclosure(1, "Великий", "Lion")
65 val enclosure2 = Enclosure(2, "Середній", "Monkey")
66 val enclosure3 = Enclosure(3, "Маленький", "Penguin")
67
```

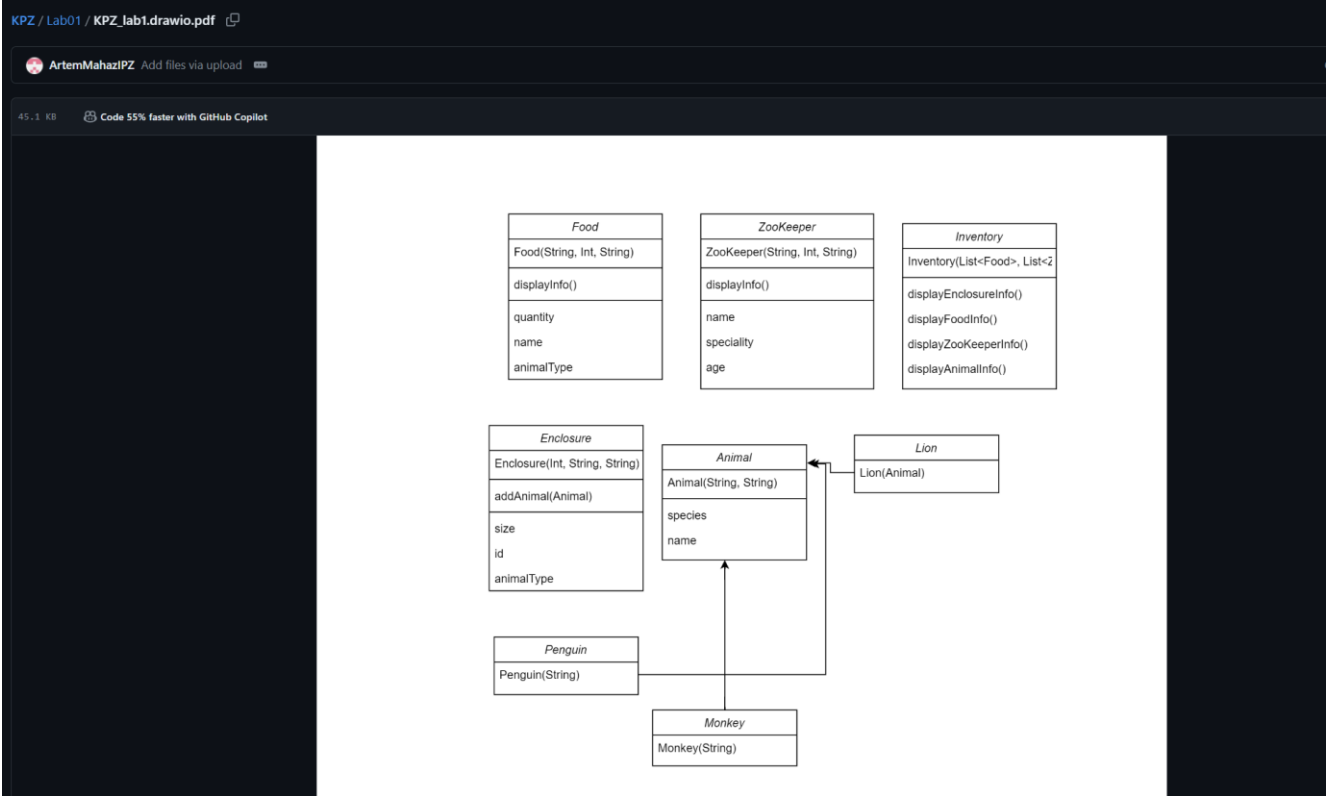
Показана легкість у створенні об'єктів класів.

Завдання 4: UML діаграма

1. Підготувати діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>.
2. Експортувати створену діаграму та запустити експортований файл у кореневу директорію цієї лабораторної.

		Магаз А. О.			ДУ «Житомирська політехніка».24.121.17.000 - Лр.1	Арк.
		Фант М. О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат:



Посилання на GitHub: <https://github.com/ArtemMahazIPZ/KPZ>