**Лабораторная работа № 7**

СОЗДАНИЕ КЛАССОВ И РАБОТА С ОБЪЕКТАМИ

**Цель работы**: получить основные понятия и навыки по созданию классов, описанию свойств и работы с объектами.

Вариант 9

**Задание:**

а) Смоделировать структуру зоопарка:

|  |  |
| --- | --- |
| Классы | Свойства |
| Зоопарк | название (get, set) |
| Вольер/клетка | номер (get, set), размер (get, set)  макс. количество животных (get, set)  текущее количество животных (get, set) |
| Животное | название (get, set)  boolean хищник (get, set) |

б) Связать между собой классы Зоопарк, Клетка и Животное, так, чтобы каждый зоопарк содержал клетки, в каждой из клеток содержатся животные, каждое животное «знает» в какой клетке оно содержится.

- Добавить в класс Зоопарк множественное свойство «клетки» (get) и методы «добавить клетку» (add), «удалить клетку» (remove).

- Добавить в класс Клетка свойство «животные» (get), а также методы «добавить животное» (add), «удалить животное» (remove). Удалить set для свойства «текущее количество животных» и сделать так, чтобы данное свойство рассчитывалось автоматически (на основе свойства «животные»).

- Добавить в класс «животное» свойство «клетка».

Создать один объект класса Зоопарк, в нем две клетки, в первую добавить два животных, во вторую одно животное. Вывести на экран животных, содержащихся в одной из клеток.

в) добавить в класс Зоопарк метод, осуществляющий поиск животного по названию. Метод содержит входной параметр Название (String) и возвращает значение типа Животное. Осуществить поиск в main(), вывести значения свойств найденного животного на экран.

**Листинг**

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Zoo { **//Класс зоопарк**

private String name;

private List<Enclosure> enclosures;

public Zoo(String name) {

this.name = name;

this.enclosures = new ArrayList<>();

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public List<Enclosure> getEnclosures() {

return enclosures;

}

public void addEnclosure(Enclosure enclosure) {

enclosures.add(enclosure);

}

public void removeEnclosure(Enclosure enclosure) {

enclosures.remove(enclosure);

}

}

class Enclosure { **//класс клетка/вольер**

private int number;

private int size;

private int maxAnimals;

private List<Animal> animals;

public Enclosure(int number, int size, int maxAnimals) {

this.number = number;

this.size = size;

this.maxAnimals = maxAnimals;

this.animals = new ArrayList<>();

}

public int getNumber() {

return number;

}

public void setNumber(int number) {

this.number = number;

}

public int getSize() {

return size;

}

public void setSize(int size) {

this.size = size;

}

public int getMaxAnimals() {

return maxAnimals;

}

public void setMaxAnimals(int maxAnimals) {

this.maxAnimals = maxAnimals;

}

public List<Animal> getAnimals() {

return animals;

}

public void addAnimal(Animal animal) {

animals.add(animal);

}

public void removeAnimal(Animal animal) {

animals.remove(animal);

}

public int getCurrentAnimals() {

return animals.size();

}

}

class Animal { **//класс животное**

private String name;

private boolean predator;

private Enclosure enclosure;

public Animal(String name, boolean predator) {

this.name = name;

this.predator = predator;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public boolean isPredator() {

return predator;

}

public void setPredator(boolean predator) {

this.predator = predator;

}

public Enclosure getEnclosure() {

return enclosure;

}

public void setEnclosure(Enclosure enclosure) {

this.enclosure = enclosure;

}

}

public class Main {

public static void main(String[] args) {

Zoo zoo = new Zoo("Центральный парк");

Enclosure enclosure1 = new Enclosure(1, 100, 10);

Enclosure enclosure2 = new Enclosure(2, 150, 15);

Animal animal1 = new Animal("Тигр", true);

Animal animal2 = new Animal("Слон", false);

Animal animal3 = new Animal("Лев", true);

zoo.addEnclosure(enclosure1);

zoo.addEnclosure(enclosure2);

enclosure1.addAnimal(animal1);

enclosure1.addAnimal(animal2);

enclosure2.addAnimal(animal3);

**//Перечисляем всех животных клетки/вольера 1**

System.out.println("Животные в вольере/клетке 1:");

for (Animal animal : enclosure1.getAnimals()) {

System.out.println("Название: " + animal.getName() + ", Жищник: " + animal.isPredator());

}

**//Перечисляем всех животных клетки/вольера 2**

System.out.println("Животные в вольере/клетке 2:");

for (Animal animal : enclosure2.getAnimals()) {

System.out.println("Название: " + animal.getName() + ", Жищник: " + animal.isPredator());

}

**//Поиск животного по названию**

System.out.println("\nПоиск животного по названию:");

Animal foundAnimal = searchAnimalByName(zoo, "Лев");

if (foundAnimal != null) {

System.out.println("Название: " + foundAnimal.getName() + ", Хищник: " + foundAnimal.isPredator());

} else {

System.out.println("Животное не найдено.");

}

**// Добавляем возможность ввода животного с клавиатуры**

Scanner scanner = new Scanner(System.in);

System.out.print("\nВведите название животного: ");

String inputName = scanner.nextLine();

System.out.print("Является ли хищником? (true/false): ");

boolean inputPredator = scanner.nextBoolean();

Animal newAnimal = new Animal(inputName, inputPredator);

System.out.println("Добавлено новое животное: " + newAnimal.getName() + ", Хищник: " + newAnimal.isPredator());

}

public static Animal searchAnimalByName(Zoo zoo, String name) {

for (Enclosure enclosure : zoo.getEnclosures()) {

for (Animal animal : enclosure.getAnimals()) {

if (animal.getName().equals(name)) {

return animal;

}

}

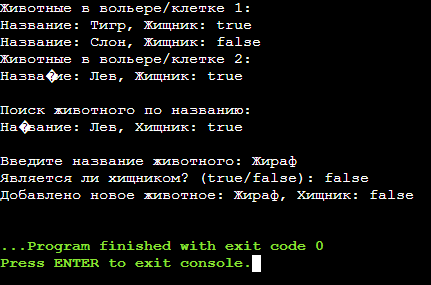
}

return null;

}

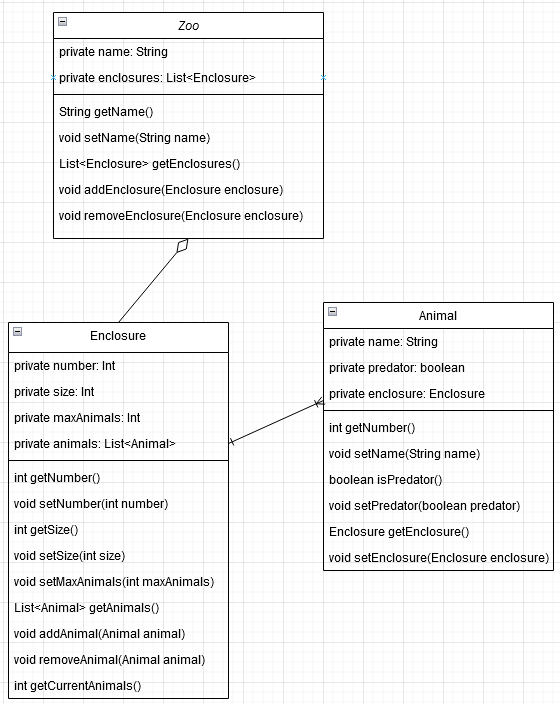
}

**Результат**



**Рисунок 1 –** Результат выполнения программы

**Диаграмма классов:**



**Рисунок 2 –** Смоделированная структура зоопарка