

Объектно-ориентированное программирование на Java

Практическая работа №5

Наследование, перечисления, обработка исключений. (9, 10 неделя)

Задание: Система управления питомцами

Описание

Создайте систему для управления питомцами в ветеринарной клинике. В системе должны быть разные типы питомцев (например, собаки и кошки), которые наследуют общие свойства от базового класса `Pet`. Кроме того, используйте перечисления для представления видов и состояний питомцев, а также добавьте обработку исключений для управления ошибками.

Задание

1. Реализуйте классы согласно описанию.
2. Добавьте возможность питомцам возвращать свои состояния (например, "здоров", "болен") с использованием перечисления `HealthStatus`.
3. Обработайте исключения, возникающие при добавлении питомца с уже существующим именем.
4. Создайте несколько питомцев и протестируйте созданные классы в главном методе.

Цели задания

- Освоить наследование и переопределение методов.
- Научиться использовать перечисления для ограниченного набора значений.

- Практиковаться в обработке исключений для улучшения надежности программ.

Классы и их структура

1. Класс `Pet` (Питомец):

- Поля:
 - `name` (имя питомца)
 - `age` (возраст питомца)
 - `type` (используйте перечисление `enum PetType`)
- Методы:
 - Конструктор для инициализации полей.
 - Метод `displayInfo`, который выводит информацию о питомце.

2. Перечисление `enum PetType` (Тип питомца):

- Значения:
 - `DOG` (Собака)
 - `CAT` (Кошка)

3. Класс `Dog` (Собака):

- Наследует от `Pet`.
- Дополнительное поле:
 - `breed` (порода собаки)
- Методы:
 - Конструктор для инициализации полей, включая `breed`.
 - Переопределите метод `displayInfo`, чтобы выводить информацию о собаке.

4. Класс `Cat` (Кошка):

- Наследует от `Pet`.
- Дополнительное поле:

- `color` (цвет кошки)
- Методы:
 - Конструктор для инициализации полей, включая `color`.
 - Переопределите метод `displayInfo`, чтобы выводить информацию о кошке.

5. Класс `PetClinic` (Ветеринарная клиника):

- Поля:
 - `pets` (список питомцев)
- Методы:
 - Конструктор для инициализации списка.
 - Метод `addPet`, который добавляет питомца в клинику.
 - Метод `displayPets`, который выводит информацию о всех питомцах.
 - Обработайте исключения, например, когда питомец уже есть в клинике.

Пример главного класса:

```
public class Main {
    public static void main(String[] args) {
        PetClinic clinic = new PetClinic();
        try {
            Dog dog1 = new Dog("Бобик", 3, "Шершавая");
            Cat cat1 = new Cat("Мурка", 2, "Черный");
            clinic.addPet(dog1);
            clinic.addPet(cat1);

            // Попытка добавить питомца с уже существующим им
            // енем
            Dog dog2 = new Dog("Бобик", 5, "Лабрадор");
            clinic.addPet(dog2); // вызовет исключение, что п
            // итomeц с таким именем уже существует
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        Cat cat2 = new Cat("Гарри", 2, "Рыжий");
        clinic.addPet(cat2);
        clinic.displayPets();
    } catch (Exception e) {
        System.out.println("Ошибка: " + e.getMessage());
    }
}
}

```

Пример вывода программы:

Имя: Бобик, Возраст: 3, Тип: DOG

Порода: Шершавая

Имя: Мурка, Возраст: 2, Тип: CAT

Цвет: Черный

Ошибка: Питомец с таким именем уже есть в клинике!

Ресурсы

1. Наследование. Преимущество наследования:
<https://javarush.com/quests/lectures/questcore.level01.lecture05>
2. Наследование: <https://metanit.com/java/tutorial/3.5.php>
3. Enum в Java: как использовать класс:
<https://javarush.com/groups/posts/1963-kak-ispoljhzovatjh-klass-enum>
4. Перечисления enum: <https://metanit.com/java/tutorial/3.8.php>
5. Исключения в Java (Java Exception):
<https://javarush.com/groups/posts/isklyucheniya-java>
6. 9 лучших практик для обработки исключений в Java:
<https://habr.com/ru/articles/551992/>