**Лабораторная работа 8: Расширение экосистемы с использованием метаклассов**

**Тема**

Модификация симуляции экосистемы (лабораторная 7) с применением **метаклассов** для гибкого управления поведением сущностей.

**Цель работы**

Научиться проектировать и использовать метаклассы в Python для динамической генерации и настройки классов растений и животных в зависимости от условий среды.

**1. Задачи**

1. **Анализ кода семестровой работы 7**
   * Ознакомиться с реализацией основных классов (Plant, Animal, производные) и механикой «тиков» симуляции.
   * Выделить повторяющиеся шаблоны создания и модификации классов (например, добавление методов обновления при смене времени суток).
2. **Проектирование метакласса EcosystemMeta**
   * Создать метакласс, который:
     + Автоматически регистрирует все подклассы растений и животных в глобальном реестре.
     + Инжектирует в класс атрибуты и методы, связанные с текущим состоянием среды (время суток, освещённость).
     + Позволяет при объявлении нового вида задавать стратегию распространения или поведение в компактном DSL‑подобном синтаксисе.
3. **Перевод существующих классов на использование метакласса**
   * Переделать определения классов Lumiere, Obscurite, Demi, Pauvre, Malheureux так, чтобы они наследовали не от object, а от базовых классов с метаклассом:

class Plant(metaclass=EvalPlantMeta): ...

class Animal(metaclass=EvalAnimalMeta): ...

* + Убрать дублирующий код «подмешивания» логики времени суток и поведения в методы классов — возложить это на метакласс.

1. **Динамическая генерация вспомогательных методов**
   * Метакласс должен автоматически создавать:
     + Для растений метод spread() с учётом конкурентных правил.
     + Для животных методы eat(), move(), reproduce(), где внутренняя логика переключается в зависимости от состояния голода, группы и времени суток.
2. **Тестирование и демонстрация**
   * Написать модуль юнит‑тестов, проверяющих:
     + Корректную регистрацию классов в реестре.
     + Присутствие автоматически сгенерированных методов.
     + Правильность поведения (spread, eat, move) в разных режимах времени.
   * Запустить демонстрацию симуляции с логом, показывающим, что изменения в метаклассе моментально отражаются на всех видах.