19. Метакласи та метапрограмування

T19.1 Абстрактний клас Drawable та його нащадок TurtleDraw Описані наступним чином:

```
class Drawable(metaclass = ABCMeta):
    """Абстрактний клас для зображення точок та кіл заданих розмірів та
кольору

    """

    @property
    @abstractmethod
    def color(self):
        """Властивість, що повертає/встановлює колір переднього плану."""
        pass

    @color.setter
    @abstractmethod
    def color(self, cl):
        pass

    @property
    @abstractmethod
    def bgcolor(self):
        """Властивість, що повертає/встановлює колір фону."""
        pass

    @bgcolor.setter
    @abstractmethod
    def bgcolor(self, cl):
        pass

    @abstractmethod
    def draw_point(self, x, y, cl):
        """Зобразити точку з координатами x, y кольором cl."""
        pass

    @abstractmethod
    def draw_circle(self, x, y, r, cl):
        """Зобразити коло з координатами цнтру x, y радіусом r кольором
cl."""
        pass


class TurtleDraw(Drawable):
    """Клас для зображення точок та кіл заданих розмірів та кольору.

    TurtleDraw є нащадком абстрактного класу Drawable та використовує засоби
    роботи з графікою з модуля turtle.
    """

    def __init__(self):
        pause = 50
        turtle.up()
        turtle.home()
        turtle.delay(pause)

    @property
    def color(self):
        """Властивість, що повертає/встановлює колір переднього плану."""
        return turtle.pencolor()
```

```
    @color.setter
    def color(self, cl):
        turtle.pencolor(cl)

    @property
    def bgcolor(self):
        """Властивість, що повертає/встановлює колір фону."""
        return turtle.bgcolor()

    @bgcolor.setter
    def bgcolor(self, cl):
        turtle.bgcolor(cl)

    def draw_point(self, x, y, cl):
        """Зобразити точку з координатами x, y кольором cl."""
        turtle.up()
        turtle.setpos(x, y)
        turtle.down()
        turtle.dot(cl)

    def draw_circle(self, x, y, r, cl):
        """Зобразити коло з координатами цнтру x, y радіусом r кольором
cl."""
        c = self.color
        self.color = cl
        turtle.up()
        turtle.setpos(x, y-r) #малює починаючи знизу кола
        turtle.down()
        turtle.circle(r)
        self.color = c
```

Додати до Drawable (та TurtleDraw) метод для зображення прямокутників. Описати клас(и) для гри у «хрестики-нолики» 3х3, який для показу поля використовує Drawable, та показати хід гри за допомогою TurtleDraw. Описати власний клас-нащадок Drawable з ім’ям CharDraw, який зображує точки, кола та прямокутники у символьному режимі. Показати хід гри за допомогою CharDraw.

T19.2 Абстрактний клас Drawable та його нащадок TurtleDraw Описані наступним чином (див. завдання T19.1).
Додати до Drawable (та TurtleDraw) метод для зображення прямокутників. Описати клас(и) для гри у «морський бій», який для показу поля використовує Drawable, та показати хід гри за допомогою TurtleDraw. Описати власний клас-нащадок Drawable з ім’ям CharDraw, який зображує точки, кола та прямокутники у символьному режимі. Показати хід гри за допомогою CharDraw.

T19.3 Описати абстрактний клас Shape (геометрична фігура) та його нащадки Circle (коло), Rectangle (прямокутник), Triangle (трикутник). Передбачити у Shape властивості обчислення периметру та площі фігури а також метод, що перевіряє, чи перетинаються дві фігури.

З використанням цих класів розв'язати задачу. Дано список геометричних фігур, що не перетинаються. Перевірити, чи справді вони не перетинаються та порахувати їх сумарну площу.

T19.4 Описати абстрактний клас Creature (істота). Кожна істота має вік, термін життя, метод зростання протягом життя, спосіб живлення (інші істоти, ґрунт тощо), Необхідний об'єм живлення, швидкість та умови розмноження. Описати класи-нащадки Creature: Animal (тварина) та Herb (рослина). Описати декілька класів тварин та рослин, задати початкову конфігурацію (кількість істот кожного класу, їх вік) та промоделювати розвиток цієї конфігурації протягом часу T.

T19.5 Описати декоратор класу, який модифікує клас для перевірки, чи належать параметри, що використовуються при ініціалізації об'єкта цього класу, заданим типам. Вважати, що клас має поле _field_types – словник, що містить імена полів в якості ключів та типи полів в якості значень. Якщо типи невідповідні, то ініціює виключення ValueError.
Застосувати цей декоратор до класів Point та Circle (див. тему «Класи та об'єкти»), та виконати зображення та переміщення точок та кіл.

T19.6 Описати декоратор класу, який здійснює модифікацію класу з метою трасування виклику усіх власних (не спеціальних) методів класу. Під час трасування показувати ім'я методу, значення параметрів до виклику, а також результат після виклику. Застосувати цей декоратор до класів Person та Student (див. тему «Класи та об'єкти») та виконати програму обчислення стипендії студентам.

T19.7 Описати декоратор класу, який здійснює модифікацію класу з метою обчислення часу роботи усіх методів класу. Під час виклику методу показувати ім'я методу та час його роботи. Застосувати цей декоратор до класу Btree (див. приклад до теми «Рекурсивні структури даних») та побудувати бінарне дерево пошуку.

T19.8 Описати декоратор класу, який здійснює модифікацію класу з метою перехоплення всіх виключень від усіх методів класу. При виникненні виключення у методі класу це виключення з усіма параметрами зберігається у текстовому файлі, а робота методу продовжується. Застосувати цей декоратор до класу Queue (див. приклад до теми «Обробка помилок та виключних ситуацій») та розв'язати наступну задачу. У магазині стоїть черга з m покупців. Час обслуговування покупця з черги – це випадкове ціле число в діапазоні від 1 до t_1. Час додавання нового покупця до черги - це випадкове ціле число в діапазоні від 1 до t_2. Промоделювати стан черги (тобто показати

час виникнення подій – обслуговування та додавання покупця) за період часу T ($T >> t_1$, $T >> t_2$). Показати залишок черги.
Вказівка: підібрати параметри m, t_1, t_2 так, щоб виникло виключення взяття елемента з порожньої черги.

T19.9 Описати декоратор класу, який здійснює модифікацію класу з метою збереження у файлі та читання з файлу значень усіх полів об'єктів класу. Для цього декоратор повинен додати у клас методи save() та load(). Застосувати цей декоратор до класу Rlist (див. приклад до теми «Рекурсивні структури даних») та реалізувати програму гри у відгадування слів зі збереженням та відновленням результатів гри.

T19.10 Описати функцію, яка створює клас-нащадок tuple, що реалізує розмічене об'єднання.
Розмічене об'єднання – це тип даних, елементи якого можуть мати різний набір полів в залежності від значення поля-дискримінанта. Наприклад, точка площини може бути представлена у декартовій або полярній системі координат. Отже, поле дискримінанта може називатись syst_coord зі значеннями 'cart', 'polar', а відповідні поля у декартовій системі координат – 'x', 'y', а у полярній, - 'ro', 'phi'. Доступ до полів розміченого об'єднання здійснюється наступним чином: a.x або a.syst_coord.
Параметрами функції, що реалізує розмічене об'єднання, будуть: ім'я класу, ім'я поля дискримінанта та словник з ключами, що є значеннями поля дискримінанта, та значеннями – наборами імен відповідних полів.
Побудований клас повинен забороняти доступ до полів, які відсутні (при заданому значенні поля дискримінанта).
Використавши цю функцію, побудувати клас для точки площини, що може бути задана у декартовій або полярній системі координат.
Нехай дано 3 точки площини у декартовій або полярній системі координат. Обчислити площу трикутника, утвореного цими точками.

T19.11 Виконати задачу T19.10 з використанням метакласів. А саме, описати метаклас UnionTupleMeta, який створює класи розміченого об'єднання та клас UnionTuple – батьківський клас у ієрархії розмічених об'єднань.

T19.12 Виконати задачу T19.9 з використанням метакласів замість декоратора. А саме, описати метаклас, який додає до створюваних ним класів save() та load().

T19.13 Виконати задачу T19.8 з використанням метакласів замість декоратора.

T19.14 Виконати задачу T19.7 з використанням метакласів замість декоратора.

T19.15 Виконати задачу T19.6 з використанням метакласів замість декоратора.

T19.16 Описати метаклас PoolMeta, який дозволяє відслідковувати, що кількість об’єктів певного класу не перевищує n. При спробі створити (n+1) об’єкт, у якості результату повертається перший створений об’єкт, далі – другий тощо.
Використати цей метаклас для розв’язання задачі. Деякий клас дозволяє зберігати та обробляти інформацію про інцидент (назва, тип, опис, час виникнення, заходи тощо). Одночасно на екрані не може розміщуватись більш, ніж 9 інцидентів. Забезпечити одночасне знаходження у пам’яті тільки 9 інцидентів (в той же час інформація про всі інциденти повинна зберігатись у текстовому файлі).