محاكاة البنية التحتية لإضاءة المدن الذكية

تطبيق أنماط هندسة البرمجيات المتقدمة (Design Patterns)

فريق العمل

نسرين مراد • رواسي خميرة • شذى جعودة

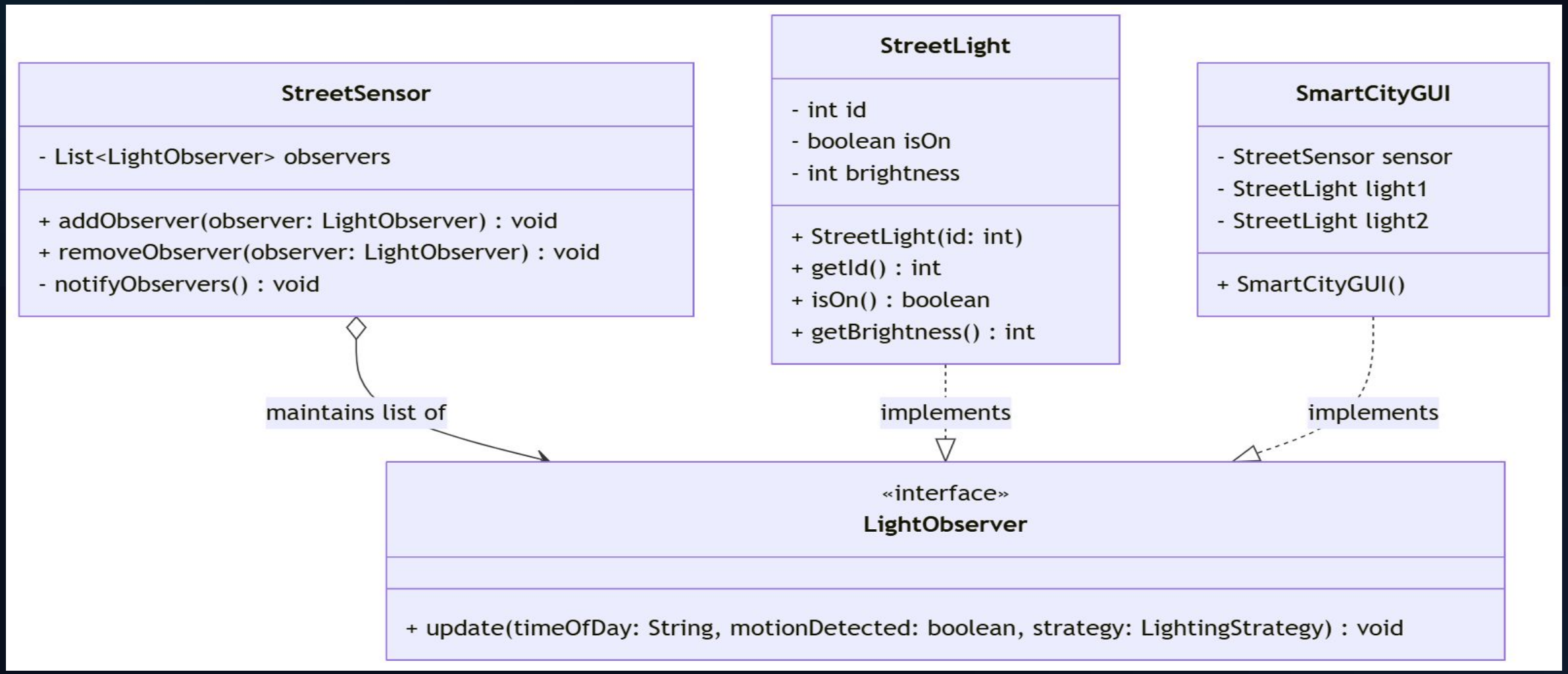
هيكلية النظام ونظرة عامة

- **المشكلة:** أعمدة الإنارة التقليدية تهدر الطاقة بالعمل بكامل طاقتها طوال الليل دون الحاجة الفعلية لذلك.
- **الحل المبتكر:** نظام محاكاة يتفاعل برمجياً (Smart City) لتعديل السطوع بناءً على الوقت ورصد الحركة في المحيط.
- **المنهجية الهندسية:**
  - فصل المنطق البرمجي (Logic) عن واجهة المستخدم (GUI).
  - تحقيق ترابط عالٍ واقتران منخفض (High Cohesion & Low Coupling).
  - الاعتماد على 3 أنماط تصميم قياسية (Design Patterns) لضمان الجودة.

نمط المراقب (Observer Pattern): البث اللحظي للأحداث

- **الهدف:** بناء علاقة (واحد إلى متعدد) لتحديث حالة المكونات في الوقت الفعلي (Real-time).
- **المُرسل (Subject):** الحساس StreetSensor يرسل التغييرات البيئية (كالحركة والوقت).
- **المستقبلون (Observers):** المصابيح StreetLight والواجهة الرسومية SmartCityGUI تستجيب تلقائياً للإشعار.
- **الفائدة الهندسية:** فك الارتباط (Decoupling) التام بين الحساس والمكونات الفيزيائية أو الرسومية، مما يسهل التطوير المستقبلي.

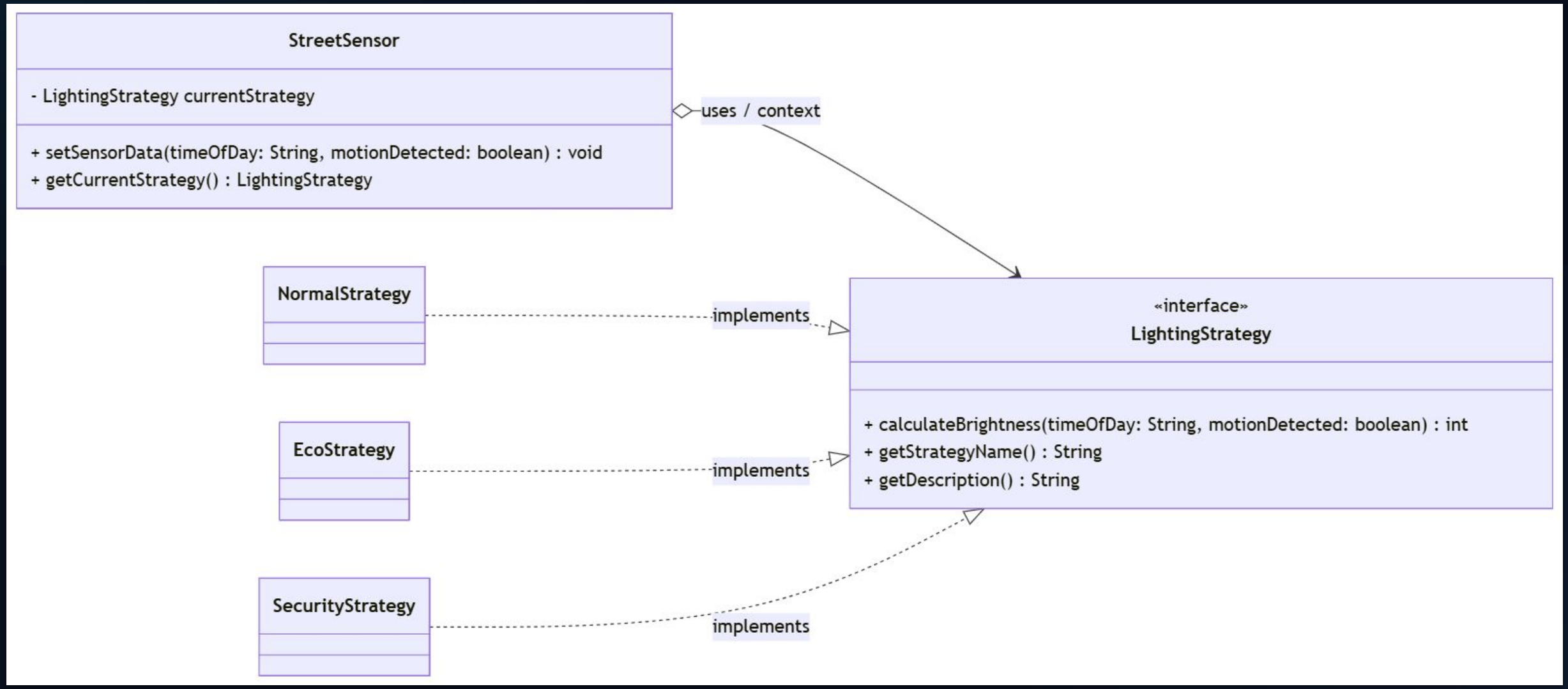
نمط المراقب (Observer Pattern): البث اللحظي للأحداث



نمط الاستراتيجية (Strategy Pattern): الخوارزميات الديناميكية

- **الهدف:** تغليف خوارزميات الإضاءة لتكون قابلة للتبديل ديناميكياً أثناء تشغيل النظام دون الحاجة لإعادة التشغيل.
- **الاستراتيجيات المطبقة (Strategies):**
  - NormalStrategy: الوضع العادي المتوازن.
  - EcoStrategy: التوفير الأقصى للطاقة.
  - SecurityStrategy: سطوع 100% مستمر للأمان.
- **الفائدة الهندسية:** تطبيق مبدأ (Open/Closed Principle)؛ حيث يمكن إضافة أوضاع جديدة دون تعديل الكود القديم أبداً.

نمط الاستراتيجية (Strategy Pattern): الخوارزميات الديناميكية



نمط المفردة (Singleton Pattern): التحكم المركزي

- **الهدف:** ضمان وجود نسخة واحدة (Instance) فقط من الكلاس يمكن الوصول إليها من أي مكان في النظام.
- **العنصر المستهدف:** الحساس المركزي للمدينة StreetSensor.
- **طريقة التطبيق:**
  - جعل المُنشئ (Constructor) من نوع private.
  - توفير دالة ثابتة للوصول static getInstance().
- **الفائدة الهندسية:** توفير (Single Source of Truth) للبنية التحتية، ومنع التضارب البرمجي الناتج عن إنشاء حساسات متعددة بالخطأ.

نمط المفردة (Singleton Pattern): التحكم المركزي

StreetSensor

- StreetSensor instance
- List<LightObserver> observers
- String timeOfDay
- boolean motionDetected
- LightingStrategy currentStrategy

- StreetSensor()
+ getInstance() : StreetSensor
+ addObserver(observer: LightObserver) : void
+ removeObserver(observer: LightObserver) : void
+ setSensorData(timeOfDay: String, motionDetected: boolean) : void
+ setSensorData(timeOfDay: String, motionDetected: boolean, strategy: LightingStrategy) : void
- notifyObservers() : void
+ getCurrentStrategy() : LightingStrategy

الخاتمة والأسئلة

نجاح النظام في محاكاة البنى التحتية الذكية بشكل احترافي.

- أنماط التصميم (Design Patterns) تقدم حلولاً جذرية وموثوقة لمشاكل البنية البرمجية.
- ساهم المشروع بشكل مباشر في تحسين قابلية الكود للصيانة والتوسع (Maintainability & Scalability).
- النظام يقدم نموذجاً مستداماً لإدارة الطاقة في المدن الذكية الحديثة.

شكراً لحسن استماعكم.

هل توجد أية أسئلة أو استفسارات؟