Correction – Étude de cas BESTLOC (GRASP)

■ Contexte

L'agence BESTLOC gère des biens immobiliers (maisons, appartements) pouvant être libres, réservés ou loués. Les clients peuvent réserver, signer un contrat ou résilier. Le directeur gère l'ajout/suppression de biens. L'agence doit aussi générer des statistiques (camemberts, histogrammes).

1■■ Diagramme de Cas d'utilisation (corrigé)

Acteurs :
- Client : réserver un bien, signer contrat, résilier contrat
- Agent : valider contrat, gérer résiliations, éditer statistiques
- Directeur : ajouter/supprimer un bien
- Banque (optionnel) : paiement dépôt

Cas d'utilisation principaux :
- Consulter biens disponibles
- Réserver un bien (création Pré-contrat)
- Payer dépôt
- Valider contrat (Agent)
- Résilier contrat
- Libérer un bien
- Ajouter/Supprimer un bien (Directeur)
- Éditer statistiques

2■■ Diagramme de Classes (corrigé et amélioré)

Classes principales :
- Bien : idBien, adresse, type, état (1=libre,2=réservé,3=loué). Méthodes réserver(), louer(), libérer().
- Client : idClient, nom, coordonnées. Méthodes réserverBien(), résilierContrat().
- PréContrat : idPréContrat, dépôt, étatValidation. Lien avec Client et Bien.
- Contrat : idContrat, dates, montant, statut. Lien avec Client et Bien.
- Directeur : ajouterBien(), supprimerBien().
- StatistiquesService : générerCamembert(), générerHistogramme(). (classe utilitaire)

3■■ Application des Patrons GRASP

- Expert → Bien gère son état (libre/réservé/loué).
- Creator → Client crée une réservation ou pré-contrat.
- Controller → ReservationController coordonne les interactions UI → domaine.
- Cohésion forte → Chaque classe reste centrée sur sa responsabilité.
- Couplage faible → Statistiques gérées dans StatistiquesService séparé.
- Pure Fabrication → StatistiquesService, classe utilitaire.
- Polymorphisme → Sous-classes de Bien (Maison, Appartement).
- Variantes protégées → État encapsulé par méthodes réserver(), louer(), libérer().