نظام لوجستيات الصيدلية والتدقيق (PHM-LOG)

🔗 رابط المخطط التفاعلي (Class Diagram):

نحن ننتقل من عالم "قواعد البيانات" (البيانات الساكنة) إلى عالم "البرمجة الكائنية OOP" (الكائنات الحية التي تتصرف وتتفاعل). في الـ Class Diagram، لدينا شيئان أساسيان:

1. الخواص (Attributes -): تمثل البيانات (ماذا يملك الكائن؟). العلامة - تعني Private (خاصة)، أي لا يمكن لأي كائن آخر تغييرها مباشرة، بل يجب أن يمر عبر أفعال الكائن نفسه لحماية البيانات.
2. الأفعال (Methods +): تمثل السلوك أو المنطق البرمجي (ماذا يفعل الكائن؟). العلامة + تعني Public (عامة)، أي يمكن للنظام استدعاؤها لتنفيذ مهمة معينة.

1. تفصيل الكائنات (Classes) والمنطق البرمجي

- ☐ كائن Patient (المريض) :
- الخواص الجديدة: لاحظ أن allergies و chronicDiseases أصبحت من نوع List<String> (قائمة من النصوص) وليس مجرد نص، لأن المريض قد يكون لديه أكثر من حساسية (بنسلين، لقاحات..).
- الأفعال (لماذا أضفناها؟):

+hasAllergy(medicationName) : هذا الفعل يسأل المريض: "هل لديك حساسية من هذا الدواء؟" يرجع بـ true او false هذا هو خط الدفاع الأول لمنع الصرف

+hasChronicDisease(diseaseName) : نفس المبدأ، يتحقق من الأمراض المزمنة لمنع التداخلات الدوائية الخطيرة.

- ☐ كائن Medication (الدواء):
- دوره: حراسة المخزون، حماية المريض من أدوية منتهية الصلاحية، وإدارة الكميات

+isExpired(): يسأل الدواء: "هل انتهت صلاحيتك؟" إذا كان true يُمنع الصرف فوراً.

+isStockSufficient(requestedQty) : يسأل الدواء: "هل الكمية المطلوبة متوفرة؟" يقارن الطلب بال stockQuantity يمنع الصرف إذا كان false

+deductStock(quantity) : الفعل الذي ينفذ طلب العميل "خصم الأدوية من المخزون بشكل مباشر".

1. يطرح الكمية المصروفة من المخزون

+checkLowStock() يتحقق إذا وصل المخزون لـ minStockAlert ليرسل تنبيه للمخازن module 7

- ☐ كائن Prescription (الوصفة الطبية)

- دوره: يمثل الحاوية (الغلاف) الذي يجمع الأدوية التي كتبها الطبيب للمريض.

+addPrescriptionItem(item) : لإضافة دواء جديد داخل هذه الوصفة.

+updateStatus(newStatus):لتغيير حالة الوصفة (مثلاً من "قيد التنفيذ" إلى "مكتملة").

▢ كائن PrescriptionItem (عنصر الوصفة)

- دوره: هذا هو "العقل المدبر" لعملية التدقيق الطبي. هو من يقرر إتمام الصرف أو رفضه بناءً على فحص المريض والدواء
- الأفعال (لماذا أضفناها؟):

+checkContraindications(patient): هذا أهم فعل في الـ module ! يأخذ كائن المريض، ويستدعي أفعال المريض patient.hasAllergy وأفعال الدواء medication.isExpired إذا وجد تعارض يرجع true

+rejectDispensing(reason) : إذا اكتشف التعارض، ينفذ هذا الفعل فيرفض الصرف ويكتب السبب في خاصية rejectionReason

+approveDispensing() : إذا كان الدواء آمناً والمخزون كافياً، يوافق على الصرف

▢ كائن DispensingLog (سجل الصرف - MAR)

- دوره: الإثبات الرسمي أن الدواء غادر الصيدلية وذهب للمريض (التدقيق)

+calculateCost() :يحسب التكلفة (الكمية المصروفة × سعر الوحدة من كائن الدواء) ويضعها في costDecimal

+sendCostToFinance():ينفذ طلب العميل "إرسال التكلفة تلقائياً للنظام المالي"، فيقوم بإنشاء بند فاتورة في الكائن التالي، ويغير خاصية isSentToFinance ل true

▢ كائن PatientBillingItem (بند الفاتورة):

- دوره: الواجهة التي تتخاطب مع النظام المالي

+processInsuranceCoverage() :يتخاطب مع نظام التأمين لمعرفة نسبة التغطية ويعدل المبلغ المتبقي على المريض.

+submitToFinanceSystem():يرسل البند النهائي ليُضاف إلى الفاتورة الكبرى للمريض في نظام المالية.

2. فهم العلاقات (لماذا ربطناها هكذا؟)

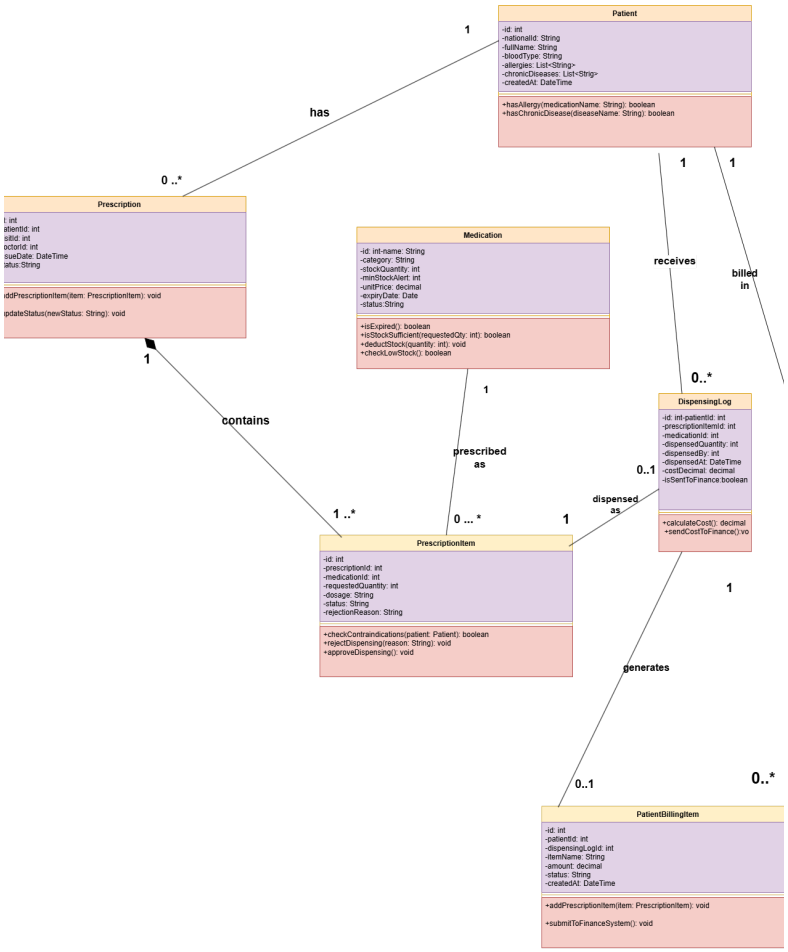
- السؤال: لماذا العلاقة بين PrescriptionItem و DispensingLog هي 1 0..1 to؟ الجواب: "لأن عنصر الوصفة (الدواء المطلوب) قد يُصرف مرة واحدة فقط (0..1)، أو قد يُرفض ولا يُصرف أبداً (0) بسبب حساسية أو نقص مخزون، لذلك لا يمكن أن يكون هناك أكثر من سجل صرف لنفس عنصر الوصفة."
- السؤال: ولماذا العلاقة بين Prescription و PrescriptionItem هي 1 1..*to؟ الجواب: "لأن الوصفة الطبية لا يمكن أن تكون فارغة أبداً. يجب أن تحتوي على دواء واحد على الأقل (1..*) لتُسمى وصفة، وإلا فلا فائدة منها."

3. السيناريو الكامل (كيف تلعب هذه الكائنات معاً؟)

لو جاء صيدلي ليصرف دواء، هكذا يعمل النظام برمجياً:

1. الصيدلي يضغط "صرف" على كائن PrescriptionItem.
2. الـ PrescriptionItem يستدعي فعله الخاص: checkContraindications(patient).
3. هذا الفعل يسأل الـ Patient: hasAllergy() ويسأل الـ Medication: isExpired() و isStockSufficient().
4. إذا كان الرد سلبي: الـ PrescriptionItem يستدعي rejectDispensing("يوجد حساسية") وينتهي الأمر.
5. إذا كان الرد إيجابي: الـ PrescriptionItem يستدعي approveDispensing().
6. ثم يطلب من الـ Medication خصم المخزون deductStock().
7. يتم إنشاء كائن DispensingLog، ويستدعي فعله calculateCost() ثم sendCostToFinance().
8. أخيراً، يُنشأ كائن PatientBillingItem ويستدعي processInsuranceCoverage() لتسجيل المبلغ على المريض.

بهذا التصميم، نحن لا نرسم أشكالاً فقط، بل نبني نظاماً ذكياً يفكر ويتحقق لحماية حياة المريض



🔗 رابط المخطط التفاعلي (Class Diagram):

NP : هذا المخطط صمم عن طريق برنامج drawo.io