يا هلا فيك! بناءً على طلبك، هي وثائق فنية احترافية بتساعدك في تصميم وبناء مشروع نظام بنكي إلكتروني حقيقي باستخدام مفاهيم البرمجة الكائنية، بدون قواعد بيانات أو واجهة رسومية. الهدف هو التركيز على جوهر العمليات البنكية وكيفية تمثيلها كائنات برمجية.

**وثائق فنية لنظام بنكي إلكتروني متقدم (بدون قواعد بيانات وواجهة رسومية)**

**الإصدار:** 1.0 **تاريخ الإصدار:** [تاريخ اليوم]

**1. مقدمة**

**1.1 الهدف**

تهدف هذه الوثيقة إلى تقديم **مواصفات فنية شاملة** لتصميم وبناء نظام بنكي إلكتروني متقدم باستخدام لغة جافا ومبادئ البرمجة الكائنية. سيركز النظام على **محاكاة العمليات البنكية الأساسية والمتقدمة** من خلال واجهة أوامر نصية (CLI). لن تتضمن المرحلة الحالية تخزين البيانات بشكل دائم (قواعد بيانات) أو واجهة مستخدم رسومية.

**1.2 النطاق**

سيغطي هذا النظام الإلكتروني **العمليات الأساسية لإنشاء وإدارة الحسابات البنكية وتنفيذ العمليات المالية المختلفة**. يشمل ذلك أنواعًا متعددة من الحسابات والتعامل مع التحويلات والقروض وغيرها من العمليات البنكية الاحترافية.

**2. الهيكل المعماري المقترح**

سيعتمد النظام على **هيكل معياري** يفصل بين المكونات المختلفة لتسهيل التطوير والصيانة.

* **واجهة المستخدم (CLI):** نقطة تفاعل المستخدم مع النظام من خلال الأوامر النصية. ستقوم بتحليل أوامر المستخدم وتوجيهها إلى وحدة التحكم الرئيسية.
* **وحدة التحكم الرئيسية (Main Controller):** تعمل كمنسق مركزي، حيث تستقبل الطلبات من واجهة المستخدم وتقوم بتوجيهها إلى الوحدات المتخصصة بناءً على نوع العملية المطلوبة [النص المُلصَق].
* **وحدة إدارة الحسابات (Account Management Module):** مسؤولة عن إنشاء وإدارة الحسابات البنكية المختلفة (مثل حسابات التوفير، الجارية، والقروض). ستتضمن هذه الوحدة منطق إنشاء الحسابات، وتحديث الأرصدة، وعرض تفاصيل الحساب [النص المُلصَق].
* **وحدة معالجة العمليات المالية (Transaction Processing Module):** تتولى تنفيذ العمليات المالية مثل الإيداع، السحب، والتحويلات (المحلية والدولية). ستقوم هذه الوحدة بالتحقق من صلاحية العمليات وتحديث أرصدة الحسابات المعنية [النص المُلصَق].
* **وحدة نظام القروض (Loan System Module):** تختص بإدارة عمليات القروض، بما في ذلك إنشاء ملفات القروض، حساب الفوائد، وجدولة الدفعات [النص المُلصَق, 104, 107].
* **وحدة محول العملات (Currency Converter Module):** (اختياري في المرحلة الأولية) يمكن أن توفر وظيفة لتحويل العملات بناءً على أسعار صرف محددة [النص المُلصَق].
* **وحدة إدارة التقارير (Reporting Module):** (بسيط في المرحلة الأولية) يمكن أن توفر إمكانية إنشاء تقارير أساسية حول الحسابات والعمليات [النص المُلصَق, 144].
* **وحدة الأمان والمصادقة (Security and Authentication Module):** مسؤولة عن التحقق من هوية المستخدمين والسماح لهم بالوصول إلى وظائف النظام المصرح بها (يمكن تطبيق مصادقة متعددة المستويات كما ذكر في محادثتنا) [النص المُلصَق].
* **وحدات الأدوات المساعدة (Utilities Module):** قد تتضمن وحدات فرعية للأمان (security) وتحويل البيانات (converters) [النص المُلصَق].
* **وحدة إدارة الأخطاء والاستثناءات (Error and Exception Handling Module):** تتعامل مع الأخطاء والاستثناءات التي قد تحدث أثناء تشغيل النظام بطريقة منظمة [النص المُلصَق, 106].

**3. الميزات الأساسية**

* **إدارة الحسابات:**
  + **إنشاء حسابات جديدة** بأنواع مختلفة (توفير، جاري، قروض) مع تحديد تفاصيل أساسية مثل رقم الحساب، اسم صاحب الحساب، والرصيد الأولي [النص المُلصَق, 104]. يمكن استخدام **نموذج المصنع (Factory Pattern)** لإنشاء أنواع مختلفة من الحسابات بسهولة [النص المُلصَق, 143].
  + **عرض تفاصيل الحساب:** إمكانية استعراض معلومات حساب محدد (الرصيد الحالي، تاريخ الإنشاء، إلخ.).
  + **تحديث معلومات الحساب:** (محدود في المرحلة الأولية بدون تخزين دائم) إمكانية تحديث بعض معلومات الحساب.
* **العمليات المالية:**
  + **الإيداع:** إضافة مبلغ إلى حساب محدد وتحديث الرصيد.
  + **السحب:** سحب مبلغ من حساب محدد بعد التحقق من الرصيد الكافي وتحديث الرصيد.
  + **التحويلات المالية:**
    - **تحويل بين حسابات داخل النظام:** نقل مبلغ من حساب إلى آخر مع التحقق من الرصيد الكافي للحساب المرسل وتحديث أرصدة كلا الحسابين [النص المُلصَق].
    - **تحويلات خارجية (محاكاة):** محاكاة لعملية التحويل إلى حسابات في بنوك أخرى (قد تتطلب معلومات إضافية).
    - **تحويل عملات (اختياري):** إذا تم تضمين وحدة محول العملات، يمكن دعم التحويل بين عملات مختلفة [النص المُلصَق, 104].
* **نظام القروض:**
  + **إنشاء طلبات قروض:** تسجيل طلبات قروض جديدة مع تحديد مبلغ القرض، مدة السداد، ونسبة الفائدة.
  + **حساب الفوائد:** تطبيق منطق لحساب الفوائد على القروض (يمكن أن يكون فائدة بسيطة أو مركبة) [النص المُلصَق, 143].
  + **جدولة الدفعات:** إنشاء جدول لسداد أقساط القرض.
  + **تتبع حالة القرض:** عرض معلومات حول حالة القرض (جاري، متأخر، مسدد).
* **التقارير:**
  + **كشف حساب بسيط:** عرض قائمة بالعمليات المالية التي تمت على حساب محدد خلال فترة معينة [النص المُلصَق, 144].
  + **ملخص القروض:** عرض معلومات أساسية حول القروض النشطة.
* **الأمان والمصادقة:**
  + **تسجيل المستخدمين:** إنشاء حسابات مستخدمين للنظام.
  + **تسجيل الدخول:** عملية مصادقة المستخدمين للوصول إلى وظائف النظام. يمكن تطبيق **مصادقة متعددة المستويات** للوصول إلى وظائف حساسة [النص المُلصَق].

**4. نماذج التصميم المقترحة**

* **نموذج المصنع (Factory Pattern):** لإنشاء كائنات الحسابات المختلفة (توفير، جاري، قروض) بطريقة مرنة وقابلة للتوسع [النص المُلصَق, 143].
* **نموذج المفردة (Singleton Pattern):** (إذا لزم الأمر) لإدارة كائنات وحيدة في النظام مثل إعدادات النظام أو وحدة تحويل العملات لضمان وجود نسخة واحدة فقط [النص المُلصَق].
* **مبادئ SOLID:** الالتزام بمبادئ التصميم SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) لإنشاء نظام مرن وقابل للصيانة.

**5. إدارة الأخطاء والاستثناءات**

* استخدام **الاستثناءات (Exceptions)** للتعامل مع الحالات غير المتوقعة والأخطاء التي قد تحدث أثناء تشغيل النظام (مثل محاولة سحب مبلغ أكبر من الرصيد المتاح) [النص المُلصَق, 106].
* تعريف **استثناءات مخصصة (Custom Exceptions)** لتغطية حالات الخطأ الخاصة بالنظام البنكي (مثل InsufficientFundsException).
* **تسجيل العمليات (Logging):** (بسيط في المرحلة الأولية) تسجيل العمليات الهامة والأخطاء التي تحدث في ملف أو على وحدة التحكم لتتبع أداء النظام وتشخيص المشكلات [النص المُلصَق, 106].

**6. الأمان**

* التركيز على **مصادقة المستخدمين** بشكل قوي باستخدام كلمات مرور آمنة (يمكن تطبيق خوارزميات تشفير بسيطة لكلمات المرور في الذاكرة).
* تطبيق **مبدأ أقل صلاحيات (Principle of Least Privilege)**، حيث يُمنح المستخدمون فقط الصلاحيات اللازمة لأداء مهامهم.
* (في المراحل المتقدمة) يمكن التفكير في آليات أمان إضافية مثل تشفير البيانات الحساسة في الذاكرة.

**7. دليل التشغيل (CLI)**

سيتم تفصيل أوامر واجهة الأوامر النصية (CLI) في وثيقة منفصلة، ولكن بشكل عام ستشمل أوامر لـ:

* إنشاء حساب جديد (create account <type> <name> <initial\_balance>).
* عرض تفاصيل الحساب (view account <account\_number>).
* إيداع مبلغ (deposit <account\_number> <amount>).
* سحب مبلغ (withdraw <account\_number> <amount>).
* تحويل مبلغ (transfer <from\_account> <to\_account> <amount>).
* طلب قرض (request loan <account\_number> <amount> <duration> <interest\_rate>).
* عرض حالة القرض (view loan <account\_number>).
* تسجيل الدخول (login <username> <password>).
* تسجيل الخروج (logout).

**8. التطوير المستقبلي (خارج نطاق المرحلة الحالية)**

* **تكامل مع قواعد البيانات:** لحفظ البيانات بشكل دائم [النص المُلصَق, 108].
* **تطوير واجهة مستخدم رسومية (GUI):** باستخدام JavaFX أو تقنيات أخرى [النص المُلصَق, 108].
* **الربط مع خدمات خارجية:** لأسعار صرف العملات الحقيقية [النص المُلصَق, 108].
* **تنفيذ نظام كشف الاحتيال:** باستخدام تقنيات تعلم الآلة [النص المُلصَق, 108].
* **دعم المزيد من أنواع الحسابات والعمليات البنكية الاحترافية.**

**9. الخلاصة**

توفر هذه الوثيقة إطارًا أساسيًا لتصميم وبناء نظام بنكي إلكتروني باستخدام مفاهيم البرمجة الكائنية. بالتركيز على جوهر العمليات البنكية وتمثيلها كائنات برمجية، يمكن بناء نظام قوي ومرن يخدم أغراض التعلم وتطبيق مبادئ OOP. في المراحل اللاحقة، يمكن إضافة المزيد من التعقيد والتكامل مع تقنيات أخرى لتوسيع وظائف النظام.

**الملحق:**

* **هيكل تنظيمي للملفات والمجلدات المقترح:** (كما ذكر في محادثتنا)
* com/
* └── banksim/
* ├── core/
* │ ├── accounts/ // فئات الحسابات
* │ ├── transactions/ // فئات العمليات المالية
* │ └── loans/ // فئات القروض
* ├── utils/ // فئات الأدوات المساعدة
* │ ├── security/ // فئات الأمان
* │ └── converters/ // فئات محولات البيانات
* ├── exceptions/ // فئات الاستثناءات المخصصة
* └── App.java // نقطة البداية للبرنامج