

مشروع: نظام إدارة مكتبة رقمية مبسط

وصف المشروع: تصميم وتطوير نظام إدارة مكتبة رقمية بسيط. يهدف النظام إلى إدارة الكتب والمستعيرين وعمليات الإعارة والاسترجاع.

الأهداف:

I. تطبيق مفاهيم البرمجة الكائنية:

- إنشاء أصناف (Classes) تمثل الكائنات الرئيسية في النظام (كتاب، مستعير، عملية إعارة).
- تطبيق مبدأ التغليف (Encapsulation) لحماية بيانات الكائنات.
- استخدام الوراثة (Inheritance) لإنشاء أصناف فرعية متخصصة (مثل كتاب ورقي، كتاب إلكتروني).
- تطبيق تعدد الأشكال (Polymorphism) للتعامل مع الكائنات المختلفة بطريقة موحدة.
- استخدام الواجهات (interface) لتحديد السلوكيات المشتركة بين الأصناف.

II. استخدام Git و GitHub

- إنشاء مستودع (Repository) على GitHub.
- إدارة إصدارات المشروع باستخدام (Git commits, branches).

III. تطوير مهارات حل المشكلات والتفكير المنطقي.

متطلبات المشروع:

1. الأصناف والكائنات:

- صنف "كتاب (Book)" يحتوي على معلومات مثل العنوان، المؤلف، رقم ISBN، الحالة (معار، متوفر) نوعه ورقي أو إلكتروني.
- صنف "مستعير (Borrower)" يحتوي على معلومات مثل الاسم، الرقم الجامعي، الكتب المستعارة.
- صنف "عملية إعارة (BorrowingProcess)" يحتوي على معلومات مثل الكتاب المستعار، المستعير، تاريخ الإعارة، تاريخ الاسترجاع.
- يمكن إضافة الأصناف حسب الحاجة

2. الوظائف:

- إضافة كتاب جديد.
- إضافة مستعير جديد.
- إعارة كتاب لمستعير.
- استرجاع كتاب من مستعير.
- البحث عن كتاب أو مستعير.
- عرض الكتب المستعارة من طرف مستعير.

3. واجهة المستخدم:

- تكون واجهة نصية (Console Application) بسيطة.

4. إدارة الإصدارات:

- يجب على الطلاب استخدام Git لإدارة إصدارات المشروع ورفعها إلى GitHub.
- يجب على الطلاب الالتزام بإنشاء commit في كل مرة يتم فيها إضافة أو تعديل خاصية أو وظيفة.

5. التوثيق:

- يجب على الطلاب توثيق الكود باستخدام التعليقات (Comments).
- يجب على الطلاب إنشاء ملف README.md يشرح كيفية تشغيل المشروع.

6. تقييم المشروع:

- جودة تصميم الأصناف وتطبيق مفاهيم البرمجة الكائنية.
- وظائف النظام ومدى تلبيتها للمتطلبات.
- جودة الكود ووضوحه.
- استخدام Git و GitHub بشكل صحيح.
- جودة التوثيق.
- عدم الاعتماد الكلي على أدوات الذكاء الاصطناعي كليا

7. طريقة تسليم الواجب:

1. إنشاء مستودع GitHub:

- يجب على كل طالب إنشاء مستودع GitHub خاص به.
- تنظيم المستودع: يجب إنشاء بنية ملفات واضحة ومنظمة داخل المستودع. على سبيل المثال:
 - مجلد "src" للكود المصدري.
 - مجلد "docs" للتوثيق (مثل ملف README.md).
 - مجلد "tests" لاختبارات الوحدة.
- ملف README.md: يجب إنشاء ملف README.md يحوي الاسم و النقب و الفوج و يشرح:
 - وصف المشروع وأهدافه.
 - كيفية تشغيل المشروع.
 - ملاحظات حول تصميم الكود وتنفيذ المفاهيم.
 - أي معلومات أخرى قد تكون مفيدة للمصحح.

2. سير عمل Git:

- (Commits):
 - يجب على الطلاب الالتزام (Commits) بالتغييرات بشكل منتظم، مع رسائل واضحة وموجزة تصف التغييرات التي تم إجراؤها.
 - يجب تجنب الالتزامات الكبيرة (Commits) التي تحتوي على العديد من التغييرات المختلفة.
 - يجب أن يكون كل commit يمثل إضافة أو تعديل لخاصية أو وظيفة.

3. طريقة التسليم و التقييم:

- رابط المستودع github:
 - إنشاء مستودع GitHub وتحميل مشروعه عليه.
 - التحقق من الرابط: يجب التأكد من أن الرابط صحيح ويعمل بشكل صحيح قبل تسليمه.
 - يجب التأكد من أن المستودع عام (Public) حتى يتمكن المصحح من الوصول إليه.
 - يجب تسليم رابط مستودع GitHub الخاص بك على الإيميل mailetudiants@gmail.com.

- الالتزام بالمواعيد: يجب الالتزام بموعد التسليم المحدد يوم **1 ماي 2025**.

- التقييم: سيتم تقييم المشاريع بناءً على:

- جودة الكود وتنظيمه.
- تطبيق مفاهيم البرمجة الكائنية.
- استخدام Git و GitHub بشكل صحيح.
- جودة التوثيق.
- الالتزام بالمعايير و المواعيد.