المشروع النهائي: نظام إدارة الطلاب والمقررات الدراسية

يقوم البرنامج بإدارة بيانات الطلاب والمقررات الدراسية باستخدام مبادئ البرمجة الأساسية، المصفوفات، البرمجة الكائنيةOOP ، الوراثة، التعامل مع النصوص Strings، والتعامل مع الملفات.

التفاصيل المطلوبة:

1. **إضافة طلاب**:
يتم إدخال بيانات الطالب التالية:
   - الاسم
   - الرقم الجامعي
   - التخصص
   - العمر
   - الطول
   - الجنسية

   عند إضافة طالب جديد، يتم تسجيل بياناته في **ملف نصي باسم studentInfo.txt** باستخدام BufferedWriterللكتابة.

2. **إضافة مقررات**:
يتم إدخال بيانات المقرر التالية:
   - اسم المقرر - كود المقرر - عدد الساعات

   عند إضافة مقرر جديد، يتم تخزين بياناته في **ملف نصي باسم course.txt** باستخدام PrintWriter للكتابة.

3. **تسجيل طالب في مقرر**:
يتم اختيار طالب من قائمة الطلاب ومقرر من قائمة المقررات لإتمام عملية التسجيل.
يتم تسجيل بيانات التسجيل الرقم الجامعي للطالب + كود المقرر في **ملف نصي باسم studentCourse.txt** باستخدام PrintWriter للكتابة.
4. **عرض بيانات الطلاب**:
يتم عرض جميع بيانات الطلاب المسجلة في الملف studentInfo.txtباستخدام BufferedReader. تتضمن البيانات: الاسم، الرقم الجامعي، التخصص، العمر، الطول، والجنسية.
5. **عرض المقررات**:
يتم عرض جميع بيانات المقررات المسجلة في الملف course.txtباستخدام

Scanner.
تتضمن البيانات: اسم المقرر، كود المقرر، وعدد الساعات.

6. **عرض بيانات الطلاب مع المقررات**:
يتم استرجاع بيانات الطلاب من الملف studentInfo.txtوبيانات التسجيل من الملف studentCourse.txtوربطهما لعرض قائمة بجميع الطلاب مع المقررات التي تم تسجيلهم فيها.
7. **حفظ واسترجاع البيانات**:
   - **حفظ البيانات** :يتم حفظ بيانات الطلاب في studentInfo.txt، وبيانات المقررات في course.txt، وبيانات التسجيل في studentCourse.txt.
   - **استرجاع البيانات** :يتم استرجاع البيانات من الملفات النصية عند تشغيل البرنامج.

المفاهيم الأساسية التي يجب استخدامها:

1) البرمجة الكائنية:OOP

- إنشاء كائنات مثل Studentو.Course
- استخدام التجميع لربط الطلاب بالمقررات.

2) الوراثة:

- إنشاء كائن Person ليكون الكائن الأب لـ .Student
يحتوي Person على البيانات التالية :
  - العمر
  - الطول
  - الجنسية
  - دالة لعرض بيانات Person الثلاثة: العمر، الطول، والجنسية.

3) التعامل مع النصوص:

- معالجة أسماء الطلاب والمقررات.

4) المصفوفات:

- تخزين بيانات الطلاب والمقررات داخليًا أثناء تشغيل البرنامج.

5) التعامل مع الملفات:

- **studentInfo.txt**:
  - تخزين بيانات الطلاب باستخدام BufferedWriterللكتابة و BufferedReaderللقراءة.

- **course.txt**:
  - تخزين بيانات المقررات باستخدام PrintWriter للكتابة و Scanner للقراءة.
- **studentCourse.txt**:
  - تخزين بيانات تسجيل الطلاب في المقررات باستخدام PrintWriter للكتابة و Scanner للقراءة.

القائمة الرئيسية:

1. إضافة طالب
2. إضافة مقرر
3. تسجيل طالب في مقرر
4. عرض بيانات الطلاب
5. عرض المقررات
6. عرض الطلاب مع المقررات
7. خروج

التقارير:

1. **عرض بيانات الطلاب**:
   يتم عرض جميع بيانات الطلاب، بما في ذلك البيانات الموروثة من Person العمر، الطول، والجنسية إلى جانب بيانات Student الاسم، الرقم الجامعي، والتخصص .
2. **عرض المقررات**:
   يتم عرض قائمة بجميع المقررات المخزنة في الملف .course.txt
3. **عرض الطلاب مع المقررات**:
   يتم عرض قائمة بجميع الطلاب والمقررات التي تم تسجيلهم فيها بناءً على البيانات في studentInfo.txtو.studentCourse.txt

---

التعليمات:

1. استخدام الوراثة لدمج البيانات المشتركة بين الطلاب ضمن الكائن .Person
2. تطبيق مبدأ التجميع Composition لربط المقررات بالطلاب.
3. التأكد من التعامل مع الملفات بطريقة صحيحة لحفظ واسترجاع البيانات.
4. تصميم واجهة تفاعلية بسيطة باستخدام القائمة الرئيسية.
5. اختبار البرنامج للتأكد من تسجيل الطلاب والمقررات بشكل صحيح وربطهم عند التسجيل.