تمرين:

* اكتب صفاً يقوم بتمثيل سجل طالب جامعي على أن يحوي هذا السجل على:
اسم الطالب - رقمه الامتحاني - علاماته - تاريخ ميلاده - تاريخ تخرجه.
* اكتب برنامجاً يقوم باستخدام الصف السابق بإدخال سجلات n طالب حيث n عدد صحيح مدخل أكبر من الواحد، ثم طباعة أسماء الطلاب مرتبة حسب:
1) تاريخ ميلادهم.
2) تاريخ تخرج كل منهم.

الحل:

إن التاريخ يحوي ثلاث معلومات (معطيات): اليوم - الشهر - السنة.
لذلك، ولتسهيل الأمر سوف نقوم قبل البدء بكتابة الصف المطلوب بتعريف صف آخر (Date) معطياته الأعضاء تعبّر عن اليوم والشهر والسنة، وسنعرّف أيضاً بداخله دالة تقوم بمقارنة تاريخين لتسهيل كتابة البرنامج المطلوب.

صف التاريخ Date:

```
class Date{
    short day;
    short month;
    short year;

    Date(){}
    Date(short d, short m, short y)
    { day=d; month=m; year=y; }

    void print(Date d){
        System.out.println(d.day+"/"+d.month+"/"+d.year);
    }
```

```
int compareTo(Date d)
{ if(year > d.year) return -1;
  else if (year < d.year) return 1;
  else if (month > d.month) return -1;
  else if (month < d.month) return 1;
  else if (day > d.day) return -1;
  else if (day < d.day) return 1;
  else return 0;
}

} // end of Date class
```

الصف الطلاب Student :

```
class Student {
  String name;
  short Deg[];
  Long Id;
  Date db;
  Date di;
  Student() { }
  Student(String name, Long Id, Date db, Date di)
  { this.name = name;
    this.Id = Id;
    this.db = db;
    this.di = di;
    Deg = new short[48];
  }
} // end of Student class
```

البرنامج المطلوب Main Student :

```
class Main Student {
  static void main (String [] s)
  { int n;
    do {n = Stdin.readInt();} while (n<=1);
    Student sa[] = new Student[n];
    for (int i=0 ; i<sa.length ; i++)
    {  String nam = Stdin.readString();
       long Id = Stdin.readLong ();
       short d = Stdin.readShort();
       short m = Stdin.read Short();
       short y = Stdin.read Short();
       Date db = new Date (d,m,y);

       Short d1 = Stdin.read Short();
       short m1 = Stdin.readShort();
       short y1 = Stdin.readShort();
       Date di = new Date (d1, m1, y1);

     Sa[i] = new Student (nam, Id, db, di);

     for (int j=0 ; j<48 ; j++)
       sa[i]. Deg[j] = Stdin.read Short();

  }
```

```
for (int i=0 ; i< sa.Length-1 ; i++)
for (int j=i+1 ; j<sa.Length    ; j++)
    if (sa[i].di.compareTo(sa[j].di) == -1)
    { Student tmp;
      tmp = sa[i];
      sa[i] = sa[j];
      sa[j] = tmp;
    }
for (int i=0 ; i< sa.Length ; i++)
{ System.out.print (sa[i].name + "____");
  sa[i].di.print (sa[i].di);
}
} // end of main
} // end of class
```

**ملاحظة هامة:**

يقوم الباني بإنشاء المطيات الأعضاء افتراضياً إذا كانت من أنواع بسيطة، ويعطيها القيم الافتراضية لهذه الأنواع إذا لم يتم تحديد القيم الابتدائية في بنية الباني نفسه ولكن إذا كانت أحد هذه المطيات الأعضاء عبارة عن كائن من نوع صف ما، فلن يستطيع الباني هنا إنشاء هذا الكائن إلا إذا كان يوجد في صف بانٍ خالٍ (افتراضي) لذا ينصح دوماً بكتابة الباني الافتراضي الخالي في الصف بحال قمنا بتعريف بانٍ واحد غيره على الأقل

مثلاً: في هذا التمرين إذا قمنا في الصف الرئيسي باستدعاء الباني غير الخالي Student وكان ضمن وسطائه المررة له كائن من النوع Date غير منشأ بعد، فلن تكون هناك مشكلة لأننا عرّفنا الباني الخالي Date () {} في الصف Date. بينما لو لم نكتب هذا الباني الخالي فلن يتمكن المترجم في هذه الحالة من إنشاء هذا الكائن تلقائياً.

## ملاحظات هامة من خلال التمرين السابق:

* كان من المفترض متى يكون البرنامج عملياً أكثر أن نحدّد شروطاً للتواريخ، فمثلاً لا يمكن لليوم أن يكون سالباً أو صفراً، كما أن الحد الأعلى لليوم يحدده الشهر والسنة، وغير ذلك من الشروط الأخرى يجب مناقشتها.

- إن هذه الشروط يجب أن توضع في دالة main وليس في إحدى دوال الصنف Date لأنه: ليس من مهمة الدالة (الطريقة) التحقق من صحة وسطائها، بل يقع على من سيستدعي الدالة التأكد من صحة الوسطاء قبل الاستدعاء.

* يكتب كل من هذه الصفوف الثلاثة في ملف مستقل كما تعلم، ومتى يستطيع أحدهم التعامل مع الصفين الآخرين فيجب أن يكون الملفان التنفيذيان (.class) لهذين الصفين في نفس المجلد الذي يوجد فيه هذا الصنف.

- وذلك لأننا لم نضع محدّدات وصول تحدّد سماحية الوصول إلى كل صنف من هذه الصفوف فيقوم المترجم بشكل افتراضي بالتعامل مع ما يسمى {"package"} أي سماحية الوصول بين الصفوف في المجلد الواحد.