**7.** معطاة الفئة **BusStation** – محطّة باص، ولها ثلاث صفات:

- num – رقم المحطّة، من نمط صحيح.
- arr – مصفوفة من نمط صحيح بكِبَر 10 ، تحوي أرقام خطوط الباصات التي تتوقّف في المحطّة. يتوقّف في المحطّة حتّى 10 خطوط، وتُحفَظ هذه الخطوط بتسلسل من بداية المصفوفة.
- amount – عدد خطوط الباصات التي تتوقّف في المحطّة بالفعل، من نمط صحيح. يتوقّف في المحطّة خطّ باص واحد على الأقلّ.

افترِضوا أنّ لصفات الفئة توجد عمليّتا get/Get وَ set/Set .

**أ.** طبِّقوا العمليّة التي أمامكم التي تتبع لواجهة تطبيق الفئة BusStation :

**Java** – public boolean isStopping (int n)

**C#** – public bool IsStopping (int n)

تتلقّى العمليّة رقم خطّ باص – n من نمط صحيح. تُعيد العمليّة true إذا كان خطّ الباص يتوقّف في المحطّة، خلاف ذلك، تُعيد false .

**ب.** معطاة مصفوفة arr من نمط BusStation ، فيها جميع محطّات الباصات في مدينة معيّنة. معلوم أنّ عددًا من خطوط الباصات تتوقّف في كلّ واحدة من المحطّات في المدينة، وبقيّة الخطوط تتوقّف فقط في جزء من المحطّات.
اكتبوا عمليّة خارجيّة باسم allStations بلغة Java أو AllStations بلغة C# ، تتلقّى المصفوفة arr .
تُعيد العمليّة سلسلة حلقات من نمط صحيح، يظهر في كلّ حلقة أحد أرقام الخطوط التي تتوقّف في **جميع** المحطّات في المدينة ( كلّ خطّ كهذا يظهر مرّة واحدة فقط في السلسلة ).
ملاحظة: لا أهمّيّة لترتيب الخطوط في السلسلة.

# الفصل الثالث

في هذا الفصل أسئلة في ثلاثة مسارات:
ألغوريثمات، في الصفحتين 12-14.
موديلات حسابيّة، في الصفحتين 15-16.
برمجة موجّهة كائنات بلغة Java ، في الصفحات 18-21؛ برمجة موجّهة كائنات بلغة #C، في الصفحات 22-25.
اختاروا أسئلة من مسار واحد فقط.

## ألغوريثمات