

معطاة الفئة **Order** ، التي تمثل تفاصيل طلبية لمُنتج معيّن، ولها أربع صفات :

- productID – الكود المميّز للمُنتج، من نمط صحيح بين 0 و 99 (بما في ذلك 0 و 99).
- price – سعر المُنتج، من نمط عدد حقيقيّ.
- quantity – كميّة وحدات المُنتج، من نمط صحيح.
- delivery – هل يشمل شحنًا إلى البيت، من نمط بوليانيّ (true – بالنسبة للطلبية التي تشمل شحنًا، false – بالنسبة للطلبية التي لا تشمل شحنًا).

الزبون الذي يطلب شحنًا إلى البيت يجب أن يدفع مبلغًا إضافيًا قدره 100 شيكل. إذا طُلبت 10 وحدات على الأقلّ، في نفس الطلبية، وبالإضافة إلى ذلك، مبلغ الطلبية (لا يشمل الشحن) هو أعلى من 170.00 شيكل – الشحن هو مجانيّ.

افترضوا أنّه توجد عمليّات get/Get و set/Set لصفات الفئة .

أ. اكتبوا في الفئة **Order** عمليّة بنائية تتلقّى الكود المميّز للمُنتج – productID وسعر المُنتج – price .

تبتدئ العمليّة صفات الفئة بحيث يتمّ تلقّي طلبية للمُنتج – productID ، بسعر المُنتج – price ، وبكميّة وحدة واحدة مع شحن إلى البيت .

ب. (1) اكتبوا في الفئة **Order** عمليّة داخلية باسم totalPrice بلغة Java أو TotalPrice بلغة C# ، تُعيد عددًا حقيقيًا للمبلغ الكليّ للدفع لجميع المنتجات في الطلبية بدون رسوم الشحن .

(2) اكتبوا في الفئة **Order** عمليّة داخلية باسم totalCost بلغة Java أو TotalCost بلغة C# ، تُعيد عددًا حقيقيًا للمبلغ يشمل رسوم الشحن، الذي يجب على الزبون أن يدفعه (إذا كانت الطلبية لا تشمل شحنًا إلى البيت أو كان الشحن مجانيًا، يُعاد مبلغ الطلبية بدون رسوم شحن) .

ملاحظة: يجب استعمال العمليّة التي كتبتموها في البند الفرعيّ "ب (1)" .

ج. معطاة مصفوفة – allOrders من نمط **Order** ، تحوي الطلبيات التي كانت في أسبوع معيّن. المصفوفة ليست مرتّبة حسب ترتيب معيّن، وليس فيها قيم null .

اكتبوا عمليّة خارجيّة باسم sumQuantity بلغة Java أو SumQuantity بلغة C# ، تتلقّى المصفوفة allOrders . تُعيد العمليّة مصفوفة جديدة من نمط صحيح بكبير 100 ، تظهر في كلّ خلية كميّة الوحدات الكليّة التي طُلبت من المُنتج، الذي الكود المميّز له مطابق لمؤشر الخلية .

مثلاً، إذا طُلبت من مُنتج الكود المميّز له هو 0 – في طلبية معيّنة 15 وحدة، وفي طلبية أخرى 20 وحدة، وفي طلبية إضافية 3 وحدات – يظهر في الخلية التي مؤشّرها 0 العدد 38 .