السؤال الأول:

إذا كانت الزيادة المحدودة هي 128 فان عدد المكالمات التي يمكننا القيام بها يتوقف على:

استخدام العملية الاتية:

عدد المكالمات= (القيمة الأقصى – القيمة الحالية) / الزيادة

السؤال الثاني:

نعم يمكن لواجهتين أن تمتدان بعضها البعض بشكل متبادل في لغة البرمجة هذا يعني أنه يمكن للواجهة الأولى ان تمتد للواجهة الثانية والعكس يمكن هذا النوع من التمديدات العديدة الاستخدامات العملية مثل تعزيز العلاقات بين الكائنات وتوسيع مجال الوظائف المتاحة للكائنات التي تنفذ في الواجهة

السؤال الثالث:

1-أحد العيوب الرئيسية هو زيادة التعقيد وصعوبة فهم النظام. عندما يكون لدينا عدد كبير من الكلاسات والتراكيب المتشعبة، يصبح من الصعب على المطورين فهم كيفية عمل النظام وكيفية التعديل عليه بشكل صحيح. قد يؤدي ذلك إلى زيادة الوقت والجهد المطلوب لإصلاح الأخطاء وتحسين النظام.

2- قد يؤدي تراكيب التوريث العميقة إلى زيادة تكرار الكود وتكرار الوظائف. عندما يكون لدينا تراكيب توريث متعددة وعميقة، فإنه قد يكون هناك حاجة لتكرار نفس الكود أو نفس الوظائف في العديد من الكلاسات المتشعبة. هذا يزيد من حجم الكود ويجعله أكثر صعوبة في الصيانة وتغييره.

3- قد يؤدي التوريث العميق إلى زيادة الوقت المستغرق في عمليات البحث والوصول إلى البيانات. عندما يكون لدينا تراكيب توريث عميقة، قد يكون هناك حاجة للعديد من العمليات الوسيطة والتحويلات للوصول إلى البيانات المطلوبة. هذا يزيد من الوقت المستغرق في تنفيذ البرنامج ويؤثر على الأداء العام للنظام..

السؤال الرابع:

1-زيادة حجم الذاكرة المستخدمة: عندما يتم تمديد العديد من الفئات من فئة واحدة، قد يؤدي ذلك إلى زيادة استهلاك الذاكرة، حيث يحتاج كل كائن من الفئات المشتقة إلى حفظ معلومات إضافية عن الفئة الأساسية.

2-زيادة في تكلفة الاستدعاءات الوظيفية: عندما يكون لديك عدة فئات تمتد من فئة واحدة، فإن استدعاء الدوال قد يستغرق وقتًا أطول، حيث يحتاج البرنامج إلى البحث في سلسلة الفئات للعثور على الدالة المناسبة.

3-زيادة في تعقيد الصيانة: عندما يكون لديك عدة فئات تمتد من فئة واحدة، فإن أي تغيير في الفئة الأساسية قد يؤثر على جميع الفئات المشتقة. هذا يعني أنه يمكن أن يكون من الصعب إجراء التغييرات والتعديلات على الفئات المشتقة بدون تأثير سلبي على البرنامج بأكمله.

لحل هذه العيوب، يمكن استخدام تقنيات أخرى مثل التركيب(composition) بدلاً من التوريث (inheritance) حيث يمكن للفئات المشتقة أن تحتوي على كائنات من الفئة الأساسية بدلاً من تمديدها. هذا يساعد في تقليل الاعتمادية على التوريث ويجعل البرنامج أكثر مرونة وسهولة في الصيانة

السؤال الخامس:

The output from calling the main () method of the Maryland class would be:

Read it.

Box it.

Buy it.

Read it.

Box it.

Buy it.

السؤال السادس:

السؤال السابع:

لا يمكن ل race ان تمتد من equestrian لأنها لا تملك علاقة معها ولكنها تمتد من horse لأنها على علاقة معها تورثها

السؤال الثامن:

try {

int [] array = new int [5];

int value = array [10];

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Don't try buffer overflow attacks in Java!");

}