##### **Задание 3. Ссылки на коллекции**

Определена иерархия классов

**class** MedicalStaff{}

**class** Doctor **extends** MedicalStaff{}

**class** Nurse **extends** MedicalStaff{}

**class** HeadDoctor **extends** Doctor{}

Укажите корректные и некорректные операторы. Дайте ответу пояснение.

Ответы помечены символом «х».

|  |  |  |
| --- | --- | --- |
|  | correct | not correct |
| Doctor doctor1 = **new** Doctor(); | x |  |
| Doctor doctor2 = **new** MedicalStaff(); |  | x |
| Doctor doctor3 = **new** HeadDoctor(); | x |  |
| Object object1 = **new** HeadDoctor(); | x |  |
| HeadDoctor doctor5 = **new** Object(); |  | x |
| Doctor doctor6 = **new** Nurse(); |  | x |
| Nurse nurse = **new** Doctor(); |  | x |
| Object object2 = **new** Nurse(); | x |  |
|  |  |  |
| List<Doctor> list1= **new** ArrayList<Doctor>(); | x |  |
| List<MedicalStaff> list2 = **new** ArrayList<Doctor>(); |  | x |
| List<Doctor> list3 = **new** ArrayList<MedicalStaff>(); |  | x |
| List<Object> list4 = **new** ArrayList<Doctor>(); |  | x |
| List<Object> list5 = **new** ArrayList<Object>(); | x |  |

Ассоциация вызова метода с телом метода называется связыванием. Для оптимизации работы Java компилятора существует разделение на «ранее(статическое) связывание» и «позднее(динамическое) связывание». «Ранним» называют связывание, которое происходит на этапе компиляции, оно относится к методам с модификаторами static, private или final. Этот тип связывания возможен, потому что такие методы не могут быть переопределены (в классах - наследниках), а тип класса можно определить в момент компиляции. «Позднее» связывание уже происходит во время работы приложения, оно и обеспечивает возможность полиморфизма в Java.

Относительно примеров, указанных в таблице, не корректными являются именно те, которые нарушают принцип «раннего связывания».