Укажите корректные и некорректные операторы. Дайте ответу пояснение.

|  |  |  |
| --- | --- | --- |
|  | correct | not correct |
| Doctor doctor1 = **new** Doctor(); | + | Создается объект соответствующий классу, никаких проблем. |
| Doctor doctor2 = **new** MedicalStaff(); | Класс MedicalStaff является классом-родителем класса Doctor. Класс «видит» только своих наследников. Ошибка при касте. | + |
| Doctor doctor3 = **new** HeadDoctor(); | + | HeadDoctor является производным классом класса Doctor, потому такое создание объекта возможно. Но, несмотря на это, будет потерян доступ к свойствам, приобретенным наследником. |
| Object object1 = **new** HeadDoctor(); | + | Ситуация, аналогичная предыдущей. |
| HeadDoctor doctor5 = **new** Object(); | Ситуация, аналогичная второй. | + |
| Doctor doctor6  = **new** Nurse(); | Пускай суперкласс и один и тот же, сами классы унаследовались параллельно, т.е. являются абсолютно различными классами. | + |
| Nurse nurse = **new** Doctor(); | Ситуация, аналогичная предыдущей. | + |
| Object object2 = **new** Nurse(); | + | Ситуация, аналогичная третьей. |
|  |  |  |
| List<Doctor> list1= **new** ArrayList<Doctor>(); | + | Все хорошо. List – это интерфейс, а ArrayList –его реализация. |
| List<MedicalStaff> list2 = **new**ArrayList<Doctor>(); | Плохо. Объект типа List должен содержать объекты типа, указанного при объявлении. Можно просто в цикле отдельно прикастить каждый элемент и уже оттуда отправлять в нужный список. | + |
| List<Doctor> list3 = **new**ArrayList<MedicalStaff>(); | Плохо. Выше написано почему. И прикастить уже нельзя. | + |
| List<Object> list4 = **new** ArrayList<Doctor>(); | Плохо. Но можно прикастить. | + |
| List<Object> list5 = **new** ArrayList<Object>(); | + | Все хорошо. |