Билет 4

# 1. Реализация исключений в Java.

**try...catch...finally**

При возникновении исключения в блоке **try** управление переходит в блок **catch**, который может обработать данное исключение. Если такого блока не найдено, то пользователю отображается сообщение о необработанном исключении, а дальнейшее выполнение программы останавливается. И чтобы подобной остановки не произошло, и надо использовать блок **try..catch**.

Конструкция **try..catch** также может иметь блок **finally**. Однако этот блок необязательный, и его можно при обработке исключений опускать. Блок **finally** выполняется в любом случае, возникло ли исключение в блоке **try** или нет.

# 2. Устаревшие коллекции в Java.

Следующие коллекции являются устаревшими, и их использование не рекомендуется, но не запрещается.

1. **Enumeration**— аналог интерфейса Iterator.  
   
2. **Vector**— аналог класса ArrayList; поддерживает упорядоченный список элементов, хранимых во "внутреннем" массиве.  
   
3. **Stack**— класс,  производный от Vector,  в который добавлены методы вталкивания (push) и выталкивания (pop) элементов,  так что список может трактоваться в терминах, принятых для описания структуры данных стека (stack).  
   
4. **Dictionary**— аналог интерфейса Map, хотя представляет собой абстрактный класс, а не интерфейс.  
   
5. **Hashtable**— аналог HashMap.

Все методы Hashtable, Stack, Vector являются синхронизированными, что делает их менее эффективными в одно поточных приложениях.