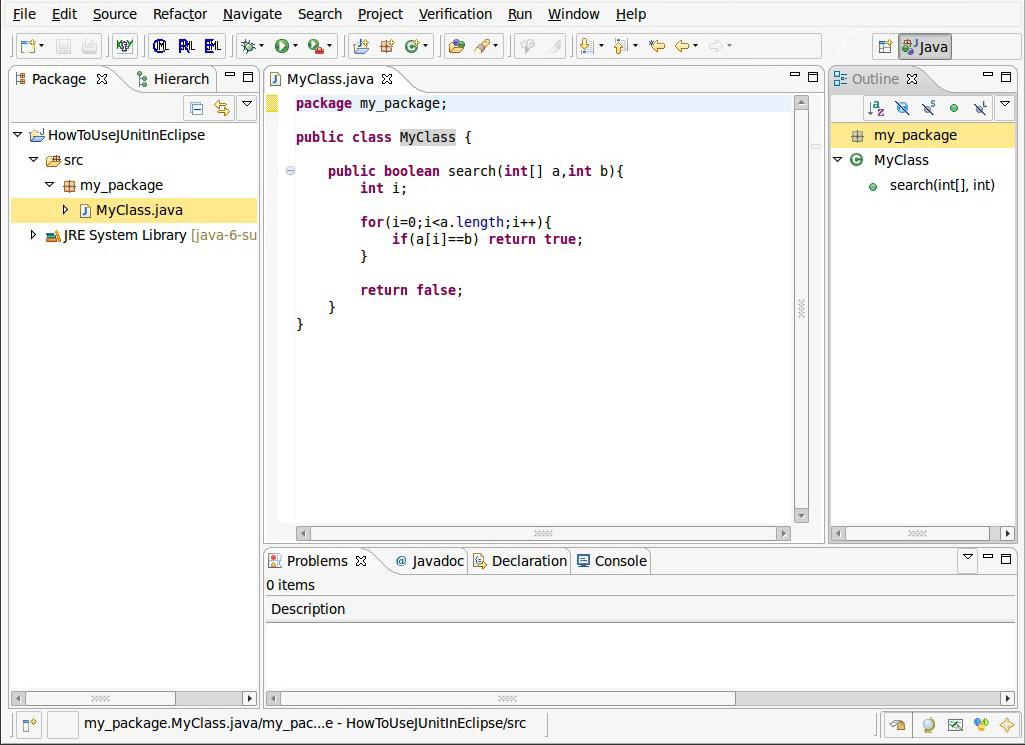
**Наги Алина ИС-31**

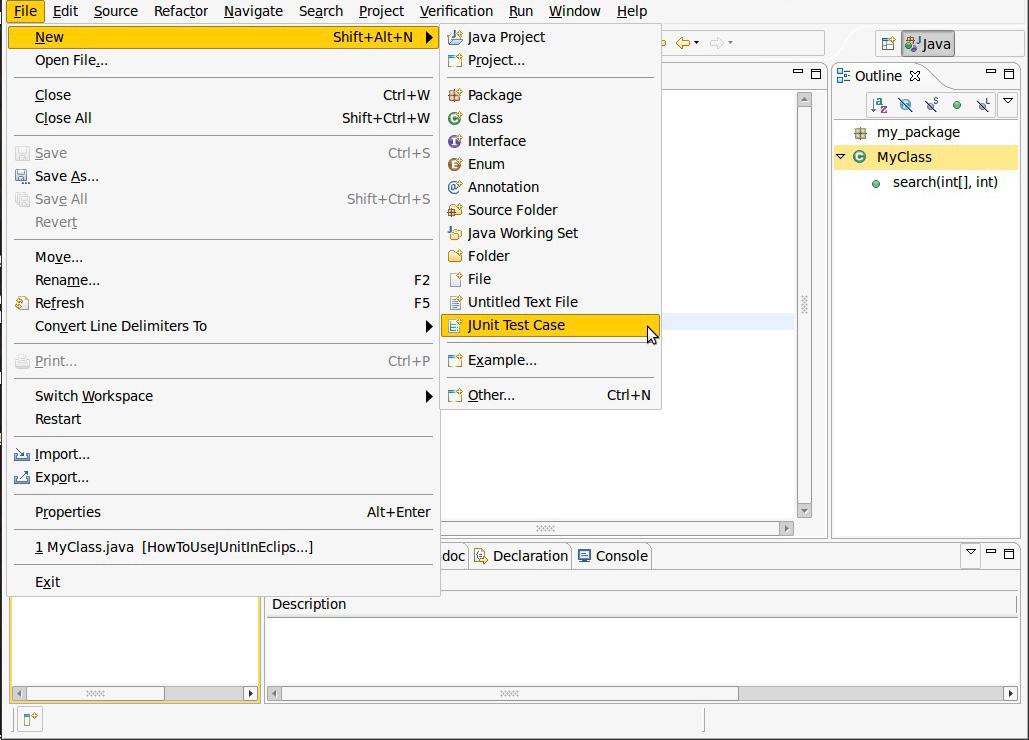
**Модульное тестирование**

**Цель работы:** 1. Изучение назначения и задач модульного тестирования.

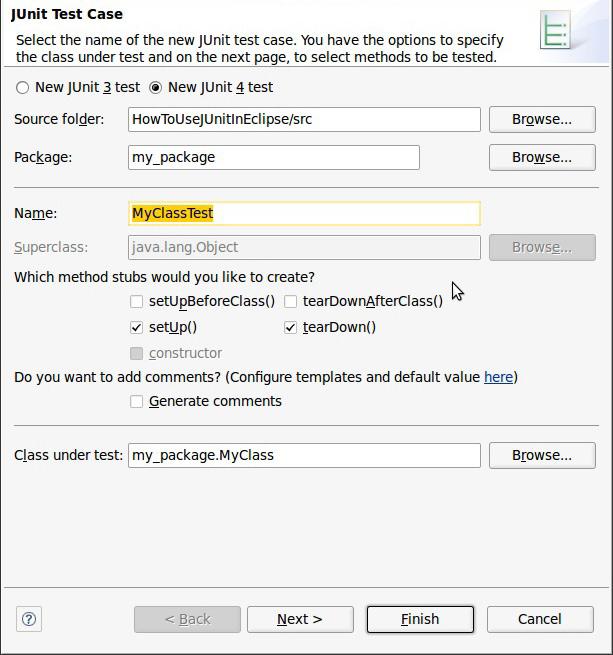
2. Программная реализация тестов, производящих модульное тестирование алгоритма пирамидальной сортировки из курса лабораторных работ.

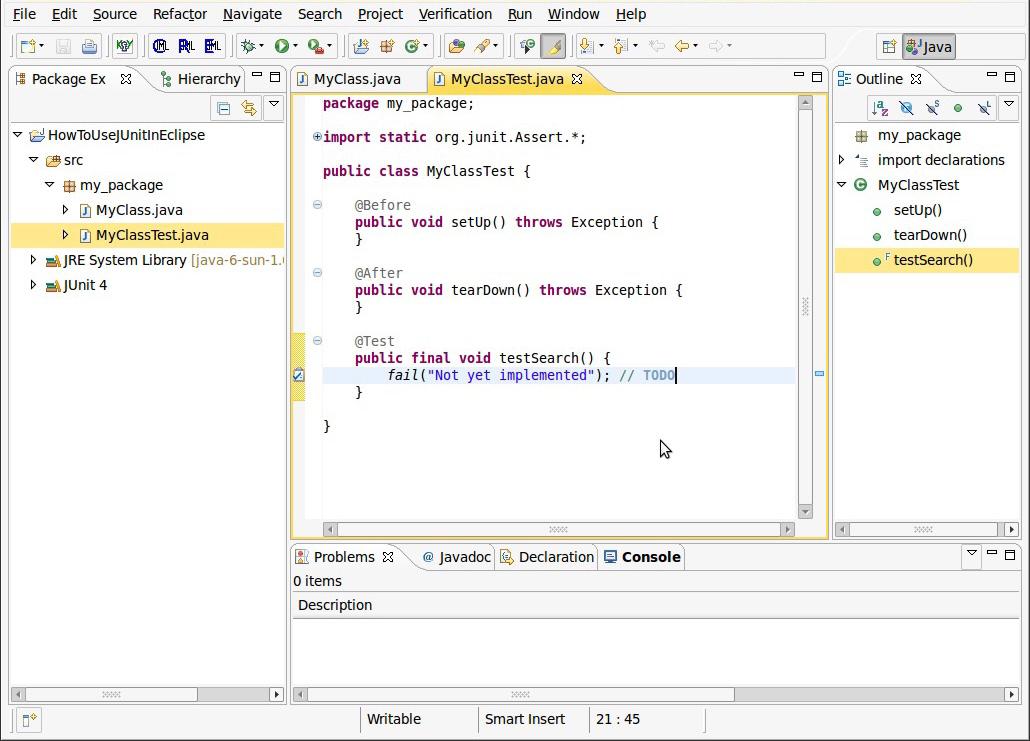


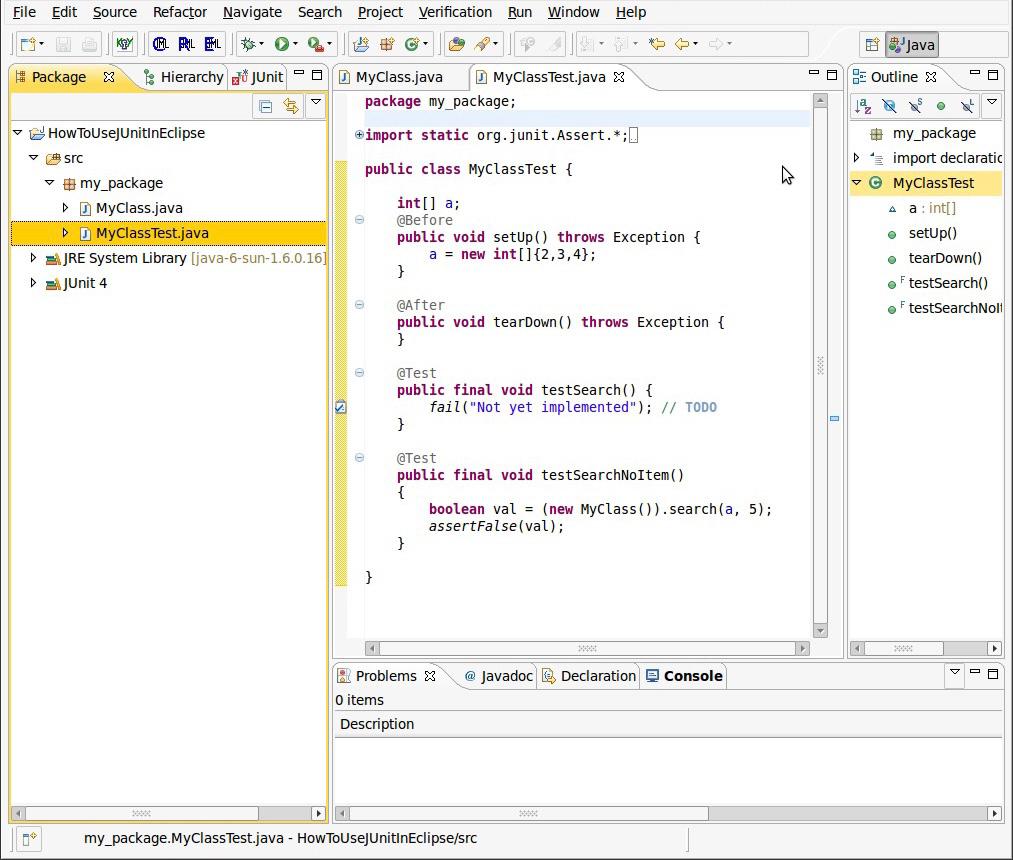
Создаю тестовый случай. В JUnit, выбрав в главном окне панели меню **Файл -**

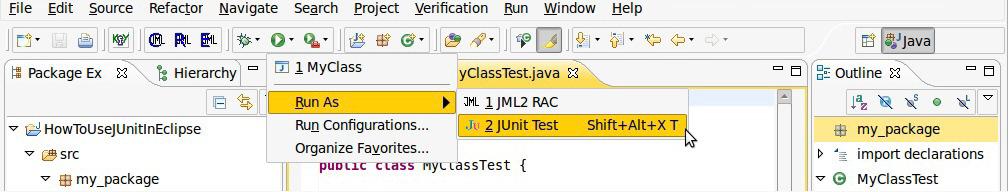


Нажатие Далее приводит к другому представлению списка чек-боксов, в котором можно выбрать методы, для которых будет генерироваться заглушки метода.

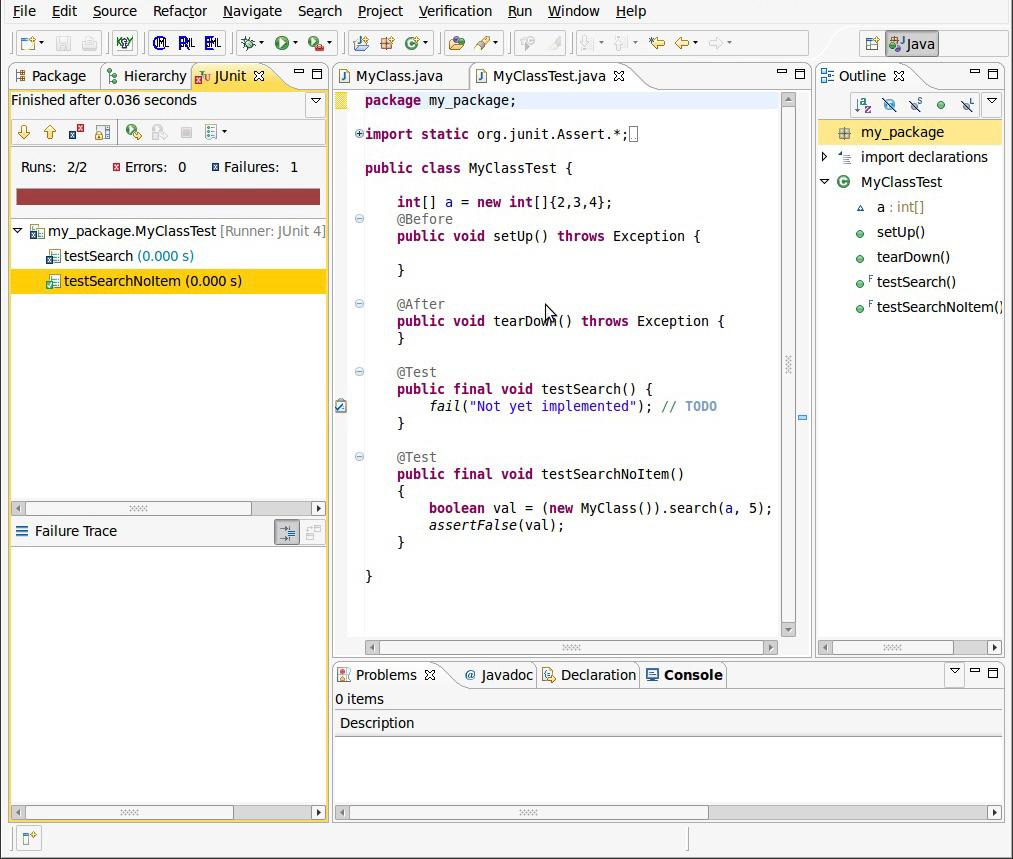








Он покажет список тестов, а также трассировку сбоя, связанную с каждым тестовым запуском.



JUnit позволяет в любой момент быстро убедиться в работоспособности кода. Если программа не является совсем простой и включает множество классов и методов, то для её проверки может потребоваться значительное время. Естественно, что данный процесс лучше автоматизировать. Использование **JUnit** позволяет проверить код программы без значительных усилий и не занимает много времени.

**Вывод:**

**Юнит тесты** классов и функций являются своего рода документацией к тому, что ожидается в результате их выполнения. И не просто документацией, а документацией которая может автоматически проверять код на соответствие предъявленным функциям. Это удобно, и часто тесты разрабатывают как вместе, так и до реализации классов. Разработка через тестирование — крайне популярная технология создания серьезного программного обеспечения.