1. **Классы как типы данных**

Цели:

* + познакомиться с модульным программированием;
	+ освоить шаблон "Неизменяемый класс";
	+ освоить модульное тестирование и JUnit в частности.

**3а. Вектор и отрезок**

Реализовать неизменяемый класс Vector3D, представляющий вектор в 3-хмерном пространстве. У класса должны быть следующие методы:

 // Получение координат

 double getX();

 double getY();

 double getZ();

 // Операции

 double scalar(Vector3D v); // скалярное произведение векторов

 double len(); // длина вектора

 Vector3D multiply(double factor); // умножение на число

 Vector3D add(Vector3D v); // сложение векторов

 Vector3D sub(Vector3D v); // вычитание векторов

Класс вектора должен быть покрыт модульными тестами, сделанными на основе библиотеки JUnit.

Затем следует реализовать класс Segment, представляющий отрезок в 3-хмерном пространстве. У класса должны быть следующие методы:

 Vector3D getStart();

 Vector3D getEnd();

 double len();

 double distanceTo(Vector3D point);

При этом в реализации методов класса Segment запрещается использование методов Vector3D.getX(), Vector3D.getY(), Vector3D.getZ().

Теперь легко заметить, что класс Segment в своей работе не использует знание о размерности пространства. В резлуьтате его легко обобщить так, чтобы он работал с векторами любого пространства. Для этого нужно:

Ввести интерфейс Vector абстрактного вектора со следующими методами

 int dimension(); // размерность

 double getComponent(int i); // возвращает компоненту вектора

 double scalar(Vector v);

 double len();

 Vector multiply(double factor);

 Vector add(Vector v);

 Vector sub(Vector v);

Указать, что класс Vector3D реализует этот интерфейс. Перевести класс Segment на использование Vector вместо Vector3D. Чтобы проверить, что Segment успешно обобщился, нужно создать реализацию вектора в 2-мерном пространстве Vector2D, и добавить соответствующих модульных тестов на класс Segment.