

Методические рекомендации к выполнению заданий

- Работы обязательно сопровождаются отчетом, включающим код программы с комментариями и экранные формы результатов выполнения программного кода.
- Студент может выполнить не все задания, а только часть в каждой работе (если их несколько). Следует отметить, что в этом случае итоговая оценка будет рассчитываться пропорционально выполненным заданиям.

Проверяемое задание 1

Тема 2. Persistence & Serialization. Персистентность и сериализация

Цель работы: сформировать навыки работы с потоками ввода-вывода Java.

Освоить технологию сериализации.

Задание. В процессе написания тестовых заданий ознакомиться с механизмом систем ввода и вывода данных.

Рекомендации по выполнению задания

1. Создайте класс `Vectors`, содержащий статические методы работы с векторами:
 - умножение вектора на скаляр;
 - сложение двух векторов;
 - нахождение скалярного произведения двух векторов.
2. Модифицируйте класс `Vectors`, добавив в него новые методы:
 - записи вектора в байтовый поток `void outputVector(Vector v, OutputStream out);`
 - чтения вектора из байтового потока `Vector inputVector(InputStream in);`
 - записи вектора в символьный поток `void writeVector(Vector v, Writer out);`

- чтения вектора из символьного потока `Vector readVector(Reader in)`.

В обоих случаях записанный вектор должен представлять собой последовательность чисел, первым из которых является размерность вектора, а остальные числа являются значениями координат вектора.

В случае символьного потока рекомендуется считать, что один вектор записывается в одну строку (числа разделены пробелами). Для чтения вектора из символьного потока рекомендуется использовать класс `StreamTokenizer`.

Проверьте возможности методов (в методе `main`), в качестве реальных потоков используя файловые потоки, а также потоки `System.in` и `System.out`.

3. Модифицируйте классы `ArrayVector` и `LinkedListVector` (основанные на массиве и на связном списке) таким образом, чтобы они были сериализуемыми.

Продемонстрируйте возможности сериализации (в методе `main`), записав в файл объект, затем считав и сравнив с исходным (по сохраненным значениям).

4. Напишите `MyClassToBePersisted.java`, который содержит следующие свойства:

- поле профиля;
- поле группы.

Напишите `SerializeMyClassToBePersisted.java`, который создает экземпляр класса `MyClassToBePersisted` и сериализует его в файл в своем основном методе.

Напишите `DeserializeMyClassToBePersisted.java`, который считывает сериализованный файл и десериализует его в экземпляр класса `MyClassToBePersisted` в своем основном методе.

5. Предоставьте отчет, который содержит текст программного кода выполненных заданий и скрины результатов работы. Приложите к отчету скомпилированные программы по выполнению указанных заданий.

При оформлении кода использование комментариев обязательно.

