**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6**

Этап 1:

Спроектировать классы указанные в задаче. Определить конструкторы, поля и методы для представления предметной области задачи. Реализовать классы в консольном приложении.

Сохранить объекты приложения в одном или нескольких файлах с применением механизма сериализации. Объекты могут содержать поля, помеченные как static, а также transient. Для извлечения информации и изменения информации в файле создать специальный классконнектор с необходимыми для выполнения этих задач методами.

Предусмотреть обработку исключений, возникающих при: нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.

Этап 2:

Модифицировать ваше решение следующим образом:

- В каждый класс предметной области добавить поле типа Date, которое должно содержать время создания объекта. Добавить печать этого поля при тестировании вашей задачи.

- Локализовать вашу задачу как минимум для следующих языков-регионов:

- русский язык, Россия

- белорусский язык, Беларусь

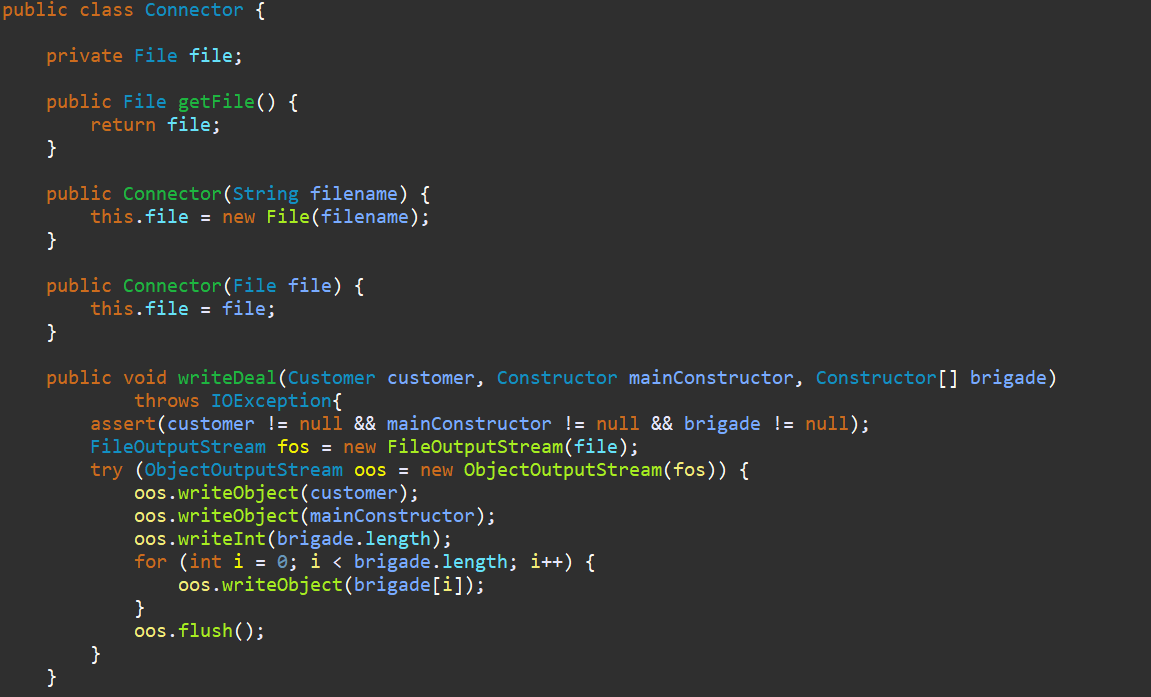
- английский язык, Великобритания

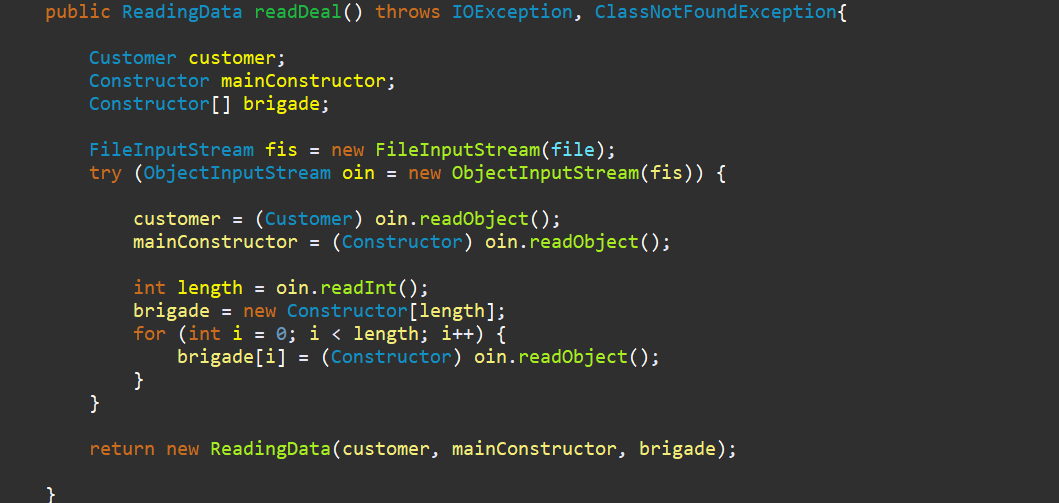
В тестовое приложение значение языка и региона передаётся с помощью параметров командной строки.

Вариант 8:

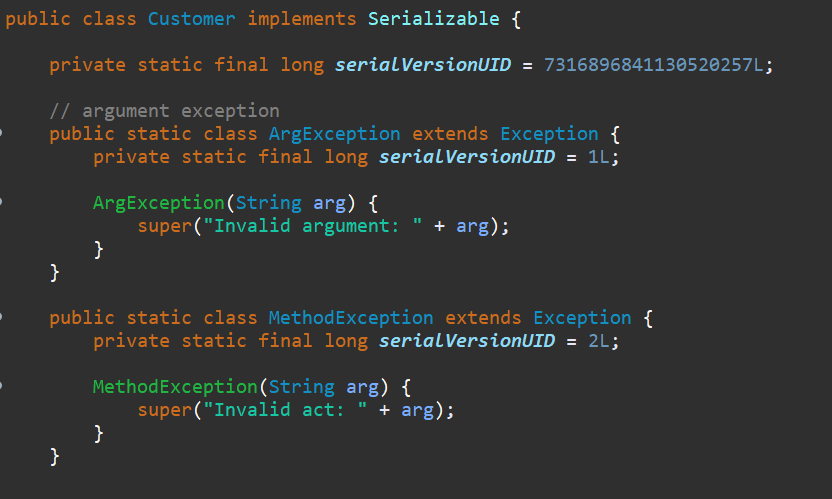
Система Конструкторское бюро. Заказчик представляет Техническое Задание (ТЗ) на проектирование многоэтажного Дома. Конструктор регистрирует ТЗ, определяет стоимость проектирования и строительства, выставляет Заказчику Счет за проектирование и создает Бригаду Конструкторов для выполнения Проекта.

ЭТАП 1:

Класс Connector хранит файл для записи/чтения данных, имеет «геттер», который возвращает файл, 2 конструктора, методы для чтения и записи данных:

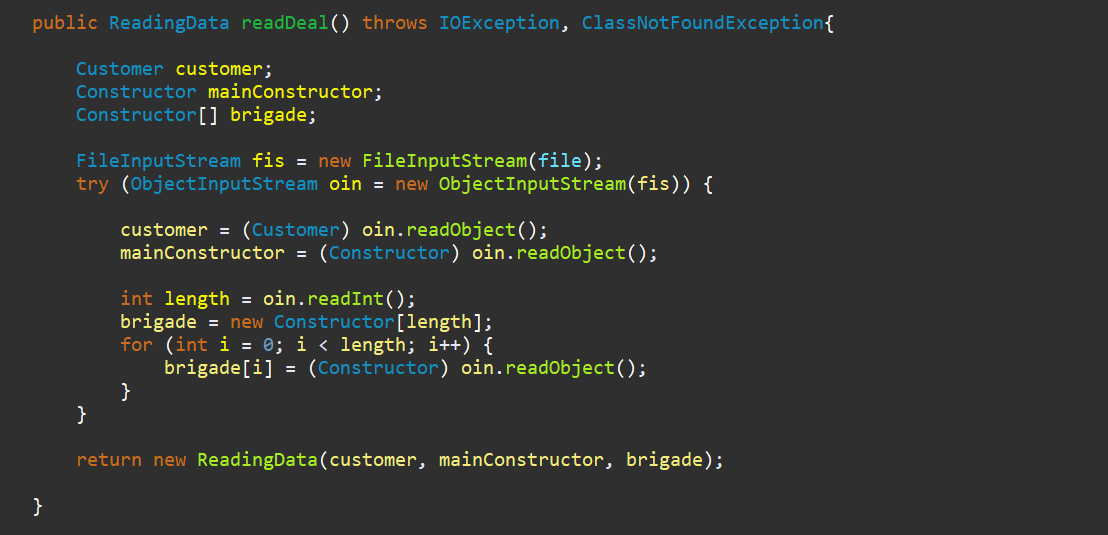


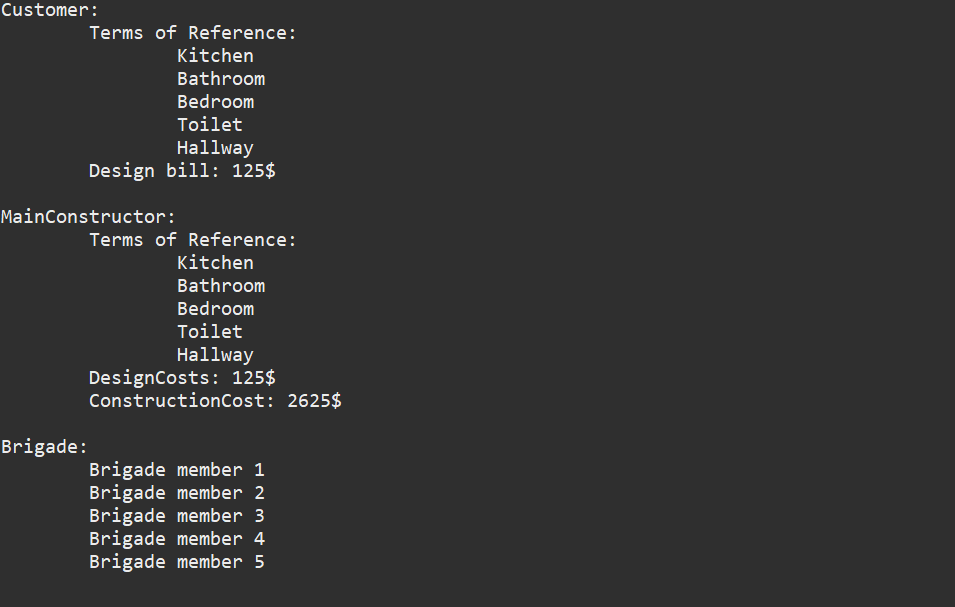
Класс Constructor хранит техническое задание, различные идентификаторы объекта, булеву переменную (является ли объект главным конструктором), конструктор для главного, приватный конструктор для членов бригады, «сеттер» для задания, «геттеры» для цен, метод создания бригады makeBrigade, метод toString.

Класс Customer хранит техническое задание и счет за проектирование, имеет классы исключений ArgException и MethodException(которые также использует класс Constructor), перечисления WorkType, «геттеры» и «сеттеры» для полей, метод toString, метод typeToString, который принимает константу из WorkType и возвращает ее строковое представление.

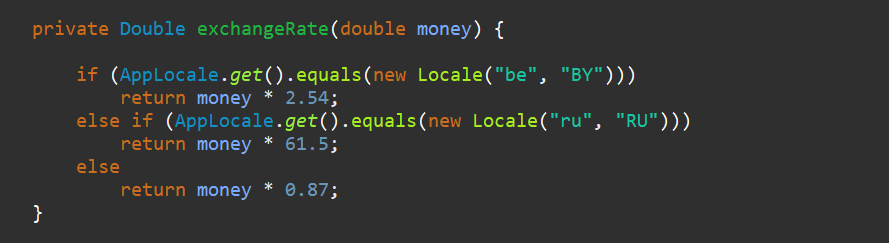
Для вычисления serialVersionUID для Customer и Constructor использовалась утилита serialver.

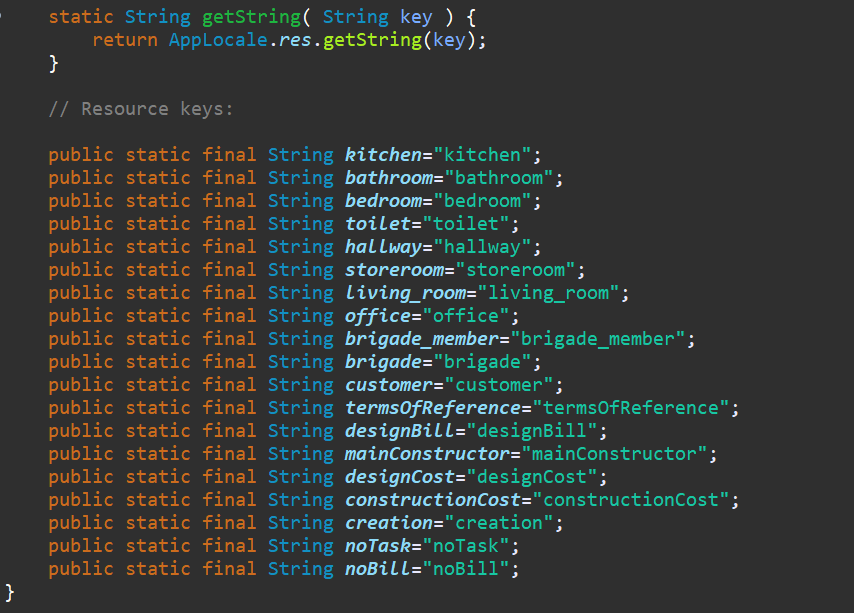
Класс ReadingData используется для возвращения нескольких объектов разных классов в методе readDeal класса Connector.

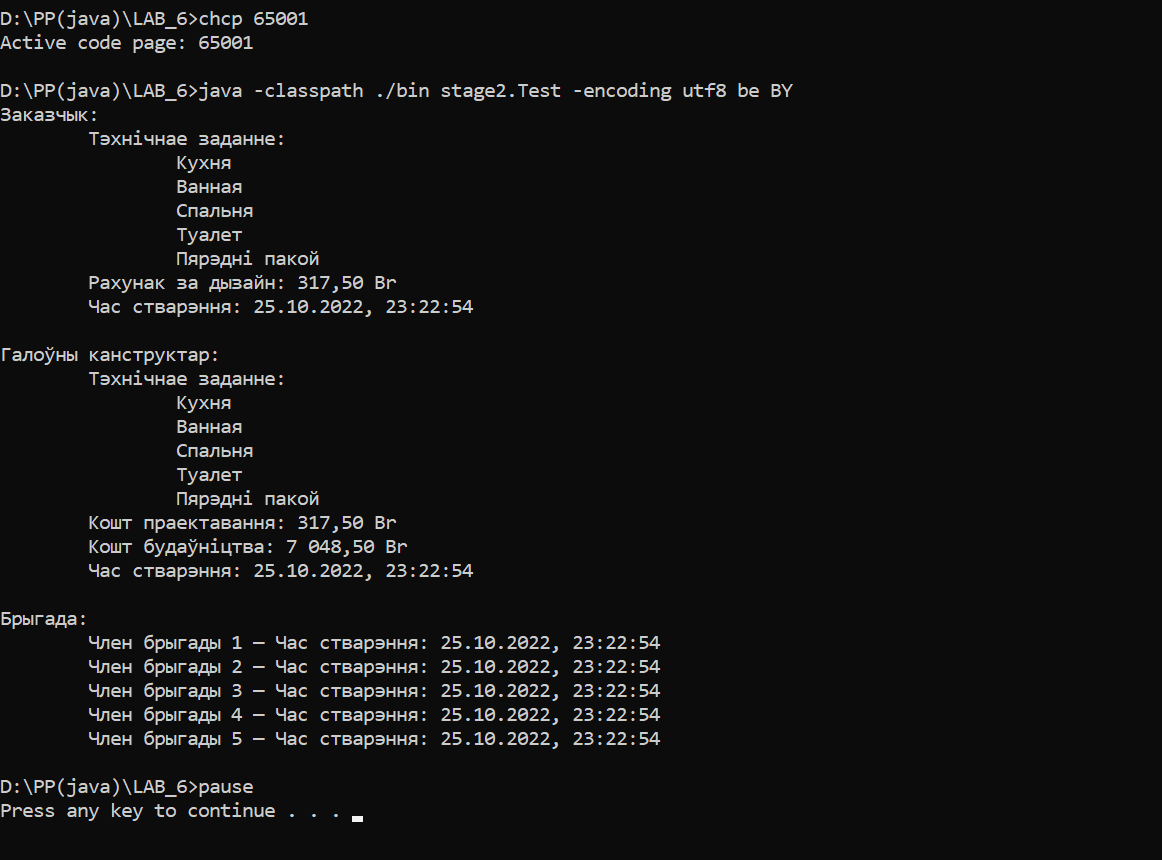


Класс Test тестирует перечисленные выше классы и их методы:

ЭТАП 2

Классы ReadingData и Connector такие же, как и в первом этапе. Классы Customer и Constructor получили локализацию в методе toString, локализацию технического задания, имеют дату создания объектов, класс Constructor теперь имеет метод exchangeRate, который переводит стоимость из долларов в нужную валюту:

Класс AppLocale хранит локализацию при помощи классов Locale и ResourceBundle и ключи из пакета ресурсов, имеет «геттеры» и «сеттеры» полей, метод getString, который возвращает определенный ключ из пакета ресурсов:

Класс Test тестирует перечисленные выше классы и их методы: