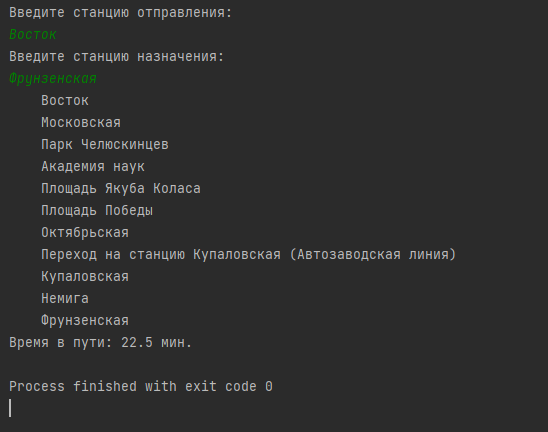
Поздравляю! Вы закончили изучение раздела Java Core. Он не такой сложный, как остальные разделы моего курса, но он очень важен. Вопросы по Core будут задавать Вам на каждом собеседовании. Поэтому знать Core очень важно для любого разработчика.  
  
Проект по Core будет квинтэссенцией всех ваших текущих знаний. Приложите все силы, вспомните все, что мы проходили.  
  
Удачи и успехов Вам!  
  
**ТЗ**:  
  
Напишите программу, которая будет строить маршруты между станциями Минского метрополитена. Пользователь вводит станцию отправления и станцию назначения, а ваша программа выводит список станций(и пересадок), а также примерное время пути (будет условно, от станции до станции 2,5 минуты).  
  
Всю информацию про минское метро ваша программа должна получать самостоятельно с сайта: [https://ru.wikipedia.org/wiki/Список\_станций\_Минского\_метрополитена](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D0%B9_%D0%9C%D0%B8%D0%BD%D1%81%D0%BA%D0%BE%D0%B3%D0%BE_%D0%BC%D0%B5%D1%82%D1%80%D0%BE%D0%BF%D0%BE%D0%BB%D0%B8%D1%82%D0%B5%D0%BD%D0%B0)  
  
**Важно**: ваша программа должна учитывать только действующие линии и станции.  
  
В вашем проекте будет 2 обязательных класса:   
**Line** - класс, содержащий информацию о линии метро (**name** - название линии, **color** - цвет линии, **stations** - список станций на данной линии);   
**Station** - класс содержащий информацию о конкретной станции (**lineName** - название линии, на которой находится станция, **name** - название станции)  
  
**Важно**: поля данных классов должны называться именно так! Отдельная сущность для переходов между станциями разных веток не нужна.  
  
Ваша программа должна сериализовывать информацию в точности как в файле minskMetroMapExample.json (приложен к тз).  
  
Ваша программа должна корректно работать когда у пользователя есть интернет (парсить информацию и сохранять ее в файл minskMetroMap.json), а также, когда интернета нет (десериализовывать информацию из ранее сохраненного файла minskMetroMap.json в готовую структуру).  
  
В программе должна быть защита от “дурака”. Например, если станции отправления или назначения нет и т.д.   
  
Для поиска маршрута вы можете использовать готовые алгоритмы (например, поиск в ширину) или придумать свой. Помните, что к одной точке может вести несколько маршрутов и вам необходимо найти самый короткий.

Для корректной сериализации и десериализации вы можете создавать любое количество любых классов. Помните про принципы ООП и чистый код. Старайтесь давать осознанные названия классам и переменным, а также, не стесняйтесь писать комментарии.

Пример работы программы вы увидите на скрине:  
  
  
  
  
Если у вас не получается печатать информацию о переходе при печати маршрута, то это не критично.

**Ход работы программы:**   
1. Вначале получаем инфу с сайта и сериализуем ее в файл по примеру (если сайт не доступен, начинаем с шага № 2)  
  
2. Затем десериализовываем файл и “собираем” метро.  
  
3. Строим маршрут.