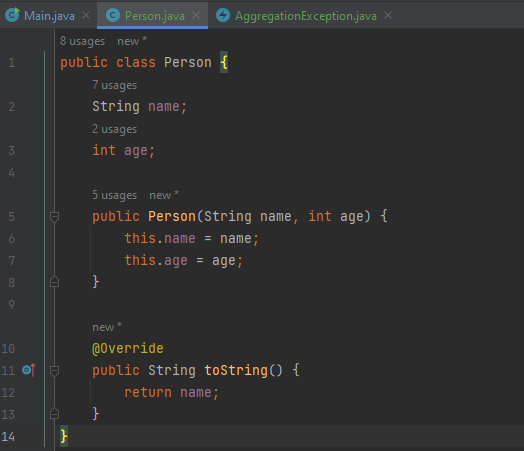
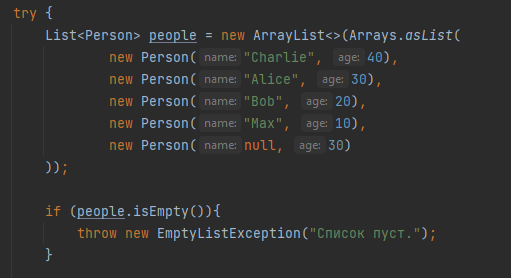
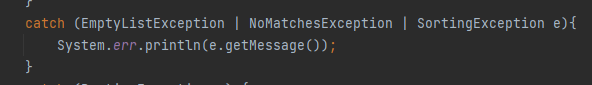
**1.** Создайте класс Person, содержащий информацию о человеке (например, имя и возраст). Затем создайте список объектов этого класса. Если список пуст, выбросите исключение EmptyListException, созданное вами. Обработайте это исключение, выводя пользователю сообщение, что список пуст.

**Создаём класс** Person c name и age атрибутами.

****

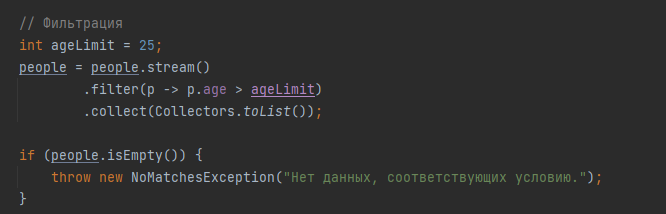
**Заполняем лист** и сразу **проверяем не пустой ли он**

Так отлавливаются некоторые исключения

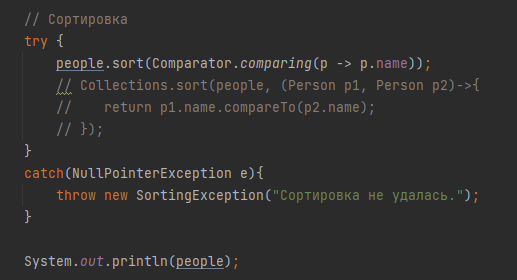
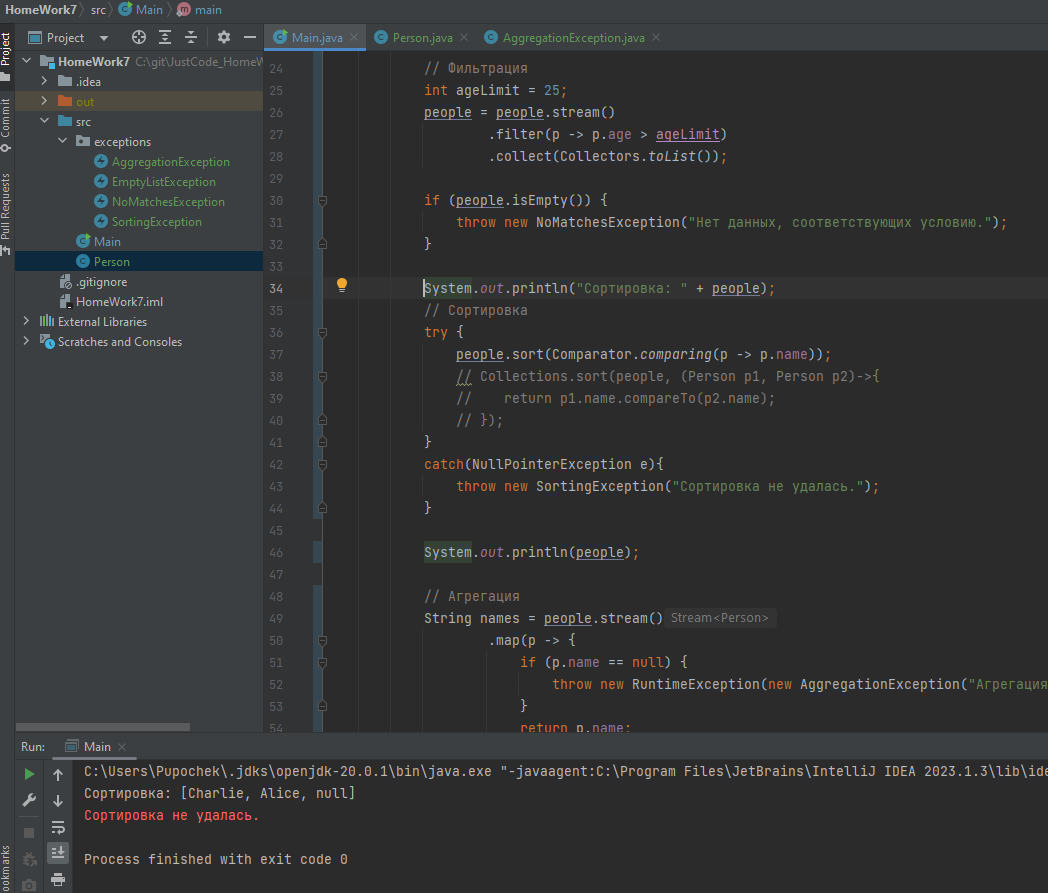


**2.** Используйте стрмы для фильтрации списка объектов Person, оставив только тех, кто старше определенного возраста. Если после фильтрации список оказывается пустым, выбросите исключение NoMatchingDataException, созданное вами. Обработайте это исключение, выводя пользователю сообщение, что нет данных, соответствующих условию.

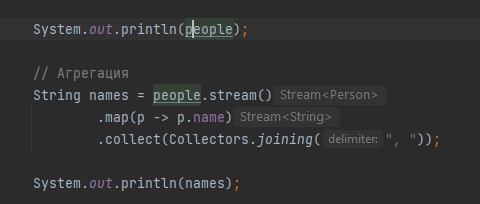
**Фильтруем** используя Stream-ы и проверяем не пустой ли вышел массив.

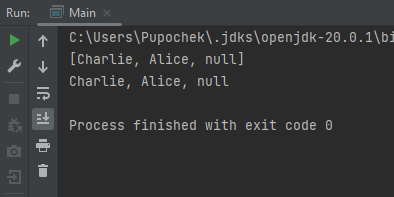


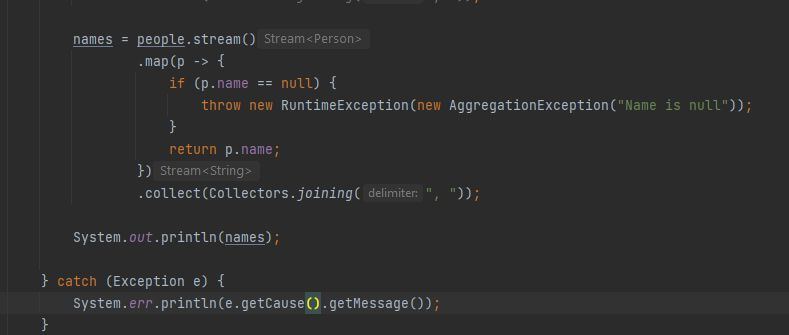
**3.** Отсортируйте список объектов Person по имени. Если во время сортировки происходит ошибка (например, из-за null-значений), выбросите исключение SortingException, созданное вами. Обработайте это исключение, выводя пользователю сообщение, что сортировка не удалась.

Здесь я написал 2 метода сортировки, каждый работает правильно. При p.name=null выходит исключение NullPointerException, которое я обертываю в SortingException.  


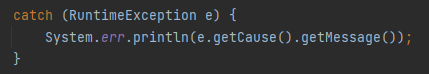
**4.** Используйте метод collect для агрегации данных, например, для создания строки со всеми именами объектов Person, разделенными запятой. Если во время агрегации происходит ошибка (например, из-за null-значений), выбросите исключение AggregationException, созданное вами. Обработайте это исключение, выводя пользователю сообщение, что агрегация данных не удалась.

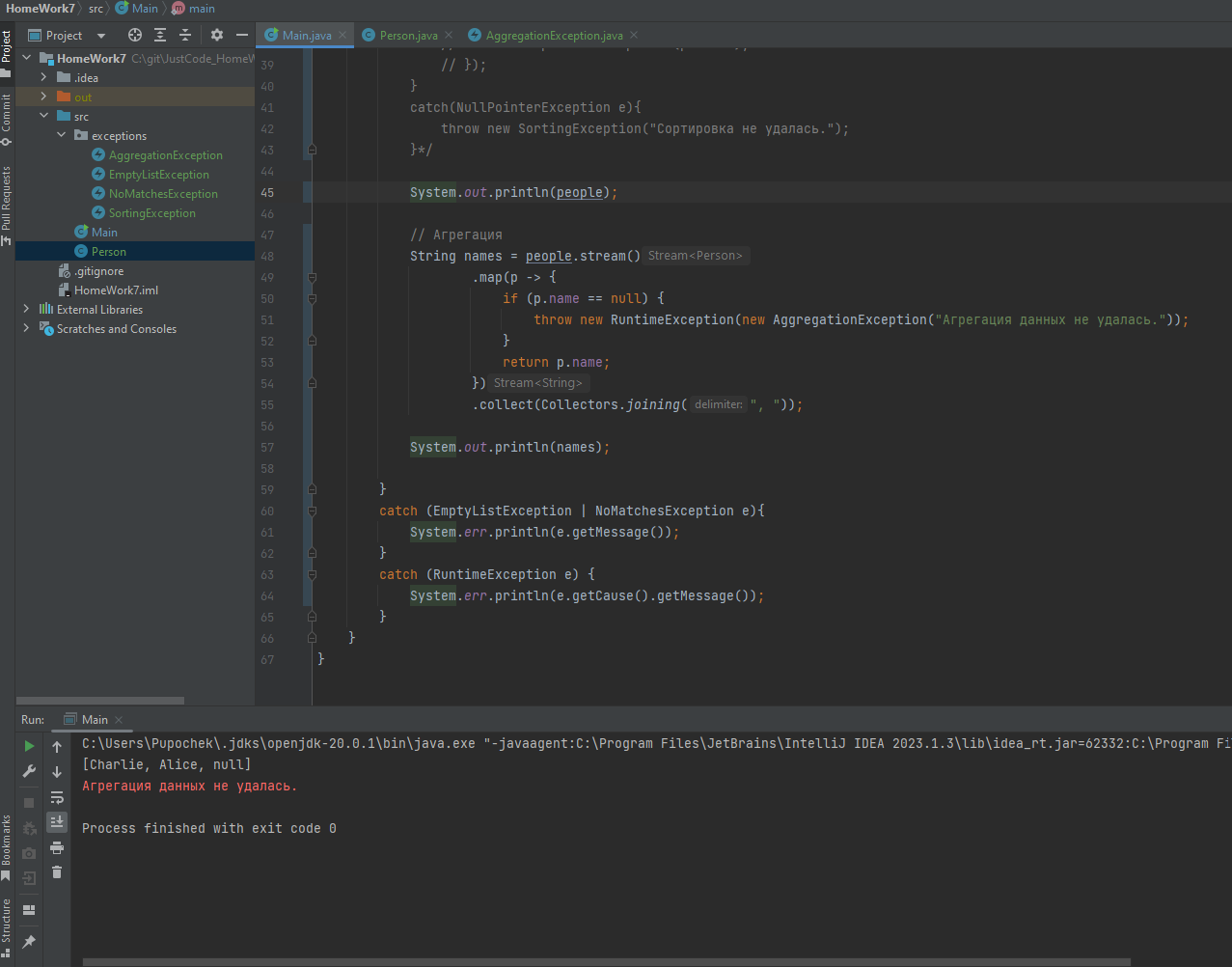
Здесь некоторые трудности с обнаружением null значения, для Collectors.joining null значение не проблема и исключение никогда не вызывается.



Поэтому я написал проверку в map. Но, из map не могут выходить checked exceptions, поэтому пришлось обернуть исключение в RuntimeException. Так делать на практике, уверен, не стоит.





Закомментировал часть с сортировкой, так как она выбросила бы исключение первой, а нам нужно **протестировать агрегацию**.   


Все созданные **исключения** хранятся в пакете exceptions и имеют такой вид:  
