

Задание на лабораторную работу

- Напишите программу, эмулирующую работу диспетчера печати документов.
- 1. Диспетчер печати может работать с несколькими типами документов (3-5 типов).
- 2. Каждый тип документа должен иметь уникальные реквизиты: продолжительность печати, наименование типа документа, размер бумаги.
- 3. Диспетчер помещает в очередь печати неограниченное количество документов. При этом каждый документ может быть обработан, только если в это же время не обрабатывается другой документ, время обработки каждого документа равно продолжительности печати данного документа.
- 4. Диспетчер должен иметь следующие методы:
 - Остановка диспетчера. Печать документов в очереди отменяется. На выходе должен быть список ненапечатанных документов.
 - Принять документ на печать. Метод не должен блокировать выполнение программы.
 - Отменить печать принятого документа, если он еще не был напечатан.
 - Получить отсортированный список напечатанных документов. Список может быть отсортирован на выбор: по порядку печати, по типу документов, по продолжительности печати, по размеру бумаги.
 - Рассчитать среднюю продолжительность печати напечатанных

Лабораторная работа – Модель банка

Необходимо разработать модель «банка» с возможностью управление через команды, вводимые в консоль.

Описание сущностей системы:

- Клиент (Customer)
 - Содержит сущности:
 - Идентификатор – строковый идентификатор в формате UUID уникально идентифицирующий клиента;
 - Имя – имя клиента;
 - Баланс – баланс;
- Транзакция (Transaction)
 - Содержит сущности:
 - Идентификатор – строковый идентификатор в формате UUID уникально идентифицирующий транзакцию;
 - Идентификатор отправителя – строковый идентификатор в формате UUID уникально идентифицирующий клиента;
 - Идентификатор получателя – строковый идентификатор в формате UUID уникально идентифицирующий клиента;
 - Сумма – сумма перевода;
 - Задержка – Величина задержки перед исполнением транзакции в условных временных единицах;
- Банк (Bank)
 - Сущности: Коллекция клиентов, очередь транзакций;
 - Методы:
 - добавление клиента: принимает объект клиента, возвращает добавленный объект;
 - удаление клиента: принимает идентификатор клиента, возвращает удаленного клиента;
 - получение отсортированных клиентов: принимает Comparator, возвращает отсортированное, с помощью полученного компаратора, множество клиентов;
 - Получение списка клиентов по имени: Принимает строку, возвращает список клиентов, имя которых содержит переданную в метод строку;
 - Получение клиента по идентификатору: Принимает строку, возвращает клиента, идентификатор которого равен переданной в метод строке;
 - Добавить транзакцию: принимает объект транзакции, возвращает true, если добавление удалось и false – если не удалось. В случае, если в транзакции указаны не существующие клиенты, выбрасывает исключение CustomerNotFoundException. Транзакции в очереди должны сортироваться по времени исполнения (задержки) от меньшего к большему;
 - Получить очередь транзакций: возвращает объект очереди транзакций.

Функциональные требования:

- Принимать команды пользователя в консоль и обрабатывать их, вызывая соответствующую функцию. В случае неизвестной команды в выходной поток ошибок консоли должно быть выведено "Unknown Command";
- Создавать, удалять, изменять данные клиентов (Customer) банка с выводом в консоль результата операции;
- Выводить в консоль отсортированный список клиентов по имени;
- Выводить в консоль отсортированный список клиентов по балансу;
- Регистрировать транзакции (Transaction) между клиентами с выводом в консоль результата операции;

- Выводить в консоль список транзакций на исполнение;
- Запускать исполнение транзакций в очереди;
- Возможность сохранить список пользователей в файл и восстановить из файла, наполнив коллекцию клиентов;
- Возможность сохранить список Транзакций в файл и восстановить из файла, наполнив очередь транзакций;

Пример

Ввод:

add-customer Bar 150

Вывод:

New Customer: id=b450403a-38fd-11e9-b210-d663bd873d93, name=Bar, balance=150

Ввод:

add-customer Foo 300

Вывод:

New Customer: id=794ad36a-38fd-11e9-b210-d663bd873d93, name=Foo, balance=300

Ввод:

print-customers-by-balance

Вывод:

Customer: id=794ad36a-38fd-11e9-b210-d663bd873d93, name=Foo, balance=300

Customer: id=b450403a-38fd-11e9-b210-d663bd873d93, name=Bar, balance=150

Ввод:

print-customers-by-name

Вывод:

Customer: id=b450403a-38fd-11e9-b210-d663bd873d93, name=Bar, balance=150

Customer: id=794ad36a-38fd-11e9-b210-d663bd873d93, name=Foo, balance=300

Ввод:

add-transaction 794ad36a-38fd-11e9-b210-d663bd873d93 b450403a-38fd-11e9-b210-d663bd873d93 100 0

Вывод:

New Transaction: id=594ad36a-23a5-12f5-b210-fad3fd908a04, from=794ad36a-38fd-11e9-b210-d663bd873d93, to=b450403a-38fd-11e9-b210-d663bd873d93, value=100, delay=0

Ввод:

print-transactions

Вывод:

Transaction: id=594ad36a-23a5-12f5-b210-fad3fd908a04, from=794ad36a-38fd-11e9-b210-d663bd873d93, to=b450403a-38fd-11e9-b210-d663bd873d93, value=100, delay=0

Ввод:

print-transactions

Вывод:

Transaction: id=594ad36a-23a5-12f5-b210-fad3fd908a04, from=794ad36a-38fd-11e9-b210-d663bd873d93, to=b450403a-38fd-11e9-b210-d663bd873d93, value=100, delay=0

Ввод:

run-transactions

Вывод:

Running Transaction: id=594ad36a-23a5-12f5-b210-fad3fd908a04, from=794ad36a-38fd-11e9-b210-d663bd873d93, to=b450403a-38fd-11e9-b210-d663bd873d93, value=100, delay=0

*Реализовать пользовательские команды с помощью шаблона проектирования «Команда»;