Примечания к отправке

Обратите внимание, что успешное прохождение общедоступных тестов необходимо для успешной подачи этого листа. Praktomat отклонит вашу заявку, если будет нарушено одно из следующих правил. Отклоненная отправка автоматически получает нулевую оценку.

• Убедитесь, что программный код компилируется без ошибок.

• Не используйте какие-либо элементы библиотек Java, за исключением элементов пакетов java.long, java.util, java.util.regex, java.util.function и java.util.stream.

Особые правила прямо упоминаются в описании задач.

• Будьте осторожны, чтобы не создавать слишком длинные строки, методы и файлы. В своих решениях вы должны придерживаться максимальной ширины строки 120 символов.

• Соблюдайте все правила использования пробелов.

• Соблюдайте все правила именования переменных, методов и пакетов.

• Выберите подходящую видимость для ваших классов, методов и атрибутов.

• Не используйте пакет по умолчанию.

• System.exit() и runtime.exit() использовать нельзя.

• Следуйте правилам документации Javadoc.

• Также соблюдайте все остальные правила Checkstyle.

Инструкции по обработке

Эти инструкции по обработке актуальны для оценки вашей заявки, но Praktomat не отклонит вашу заявку, если одно из следующих правил будет нарушено.

• Обратите внимание, что ваши заявки будут оцениваться как с точки зрения объектно-ориентированного моделирования, так и с точки зрения функциональности. Следуйте инструкциям по моделированию в Ilias Wiki.

• Программный код должен быть написан на английском языке.

• Правильно комментируйте свой код: как можно больше, как можно меньше.

• Комментарии должны быть написаны на английском или немецком языке.

• Выберите понятные имена для всех своих идентификаторов.

Плагиат

Принимаются только самодельные решения. Отправка сторонних решений, даже если только частичные решения от третьих лиц, из книг, Интернета или других источников, является попыткой обмана и приводит к оценке «не удалось». Исключительно отрывки исходного текста из слайдов лекций и из предлагаемых решений из практики в этом семестре. Все используемые вспомогательные средства должны быть указаны полностью и точно, и все, что было взято из работы других без изменений или с изменениями, должно быть четко идентифицировано. Точно так же передача решения или его частей представляет собой нарушение упорядоченного процесса контроля успеха. Это нарушение правил также может привести к исключению контроля успеха. Для получения дополнительной информации обратитесь к заявлению о согласии.

Checkstyle

Praktomat автоматически проверяет исходные тексты при отправке на соответствие правилам Checkstyle. Существуют специально обозначенные правила, по которым Praktomat отклоняет подачу заявки, так как это правило является обязательным. Другие нарушения правил могут привести к вычету баллов. Вы можете и должны проверять исходные тексты на соответствие правилам во время разработки.

Класс терминала

Для этой задачи загрузите класс терминала и обязательно включите его в пакет edu.kit .informatik. Метод Terminal.readLine () считывает ввод пользователя с консоли и заменяет System.in. Метод Terminal.printLine () записывает вывод в консоль и заменяет System .out. Используйте класс Terminal для любого ввода или вывода консоли. Никогда не используйте System.in или System.out. Никогда не загружайте класс терминала вместе с отправкой.

Задача B: Почтовое отделение.

Целью данной задачи является внедрение системы для почтового отделения, в которой могут регистрироваться разные пользователи. Для регистрации требуется удостоверение личности. После успешной регистрации существующие пользователи могут пользоваться почтовыми услугами, такими как получение и отправка посылки. **Клиент** может войти в систему со своим личным именем пользователя и паролем.

Почтовый работник может войти в систему почтового отделения, используя свой персональный номер и пароль. В зарегистрированном состоянии он может отправлять почту от имени клиента и передавать клиенту его сохраненную почту. Почтовый работник также может предоставить клиенту дополнительную информацию об уже предоставленных почтовых услугах (см. List - mail command и list - price command).

Сотрудник колл-центра также может войти в систему почтового отделения по личному номеру и паролю. Подобно почтовому работнику, он может предоставить клиентам информацию об их аккаунте в почтовом отделении и сбросить пароль клиента (см. Команду reset-pin).

Есть несколько **почтовых служб**, которые клиент может отправлять и получать. Почтовые услуги имеют разные цены. Письма стоят 0,70 денежных единиц, заказная почта - 1,20 денежных единиц, заказная почта - 2,00 денежных единицы.(в евро) Есть разные размеры для пакетов с разной ценой. Стоимость PaketS составляет 5,00, PaketM - 6,00, а PaketL - 7,00 денежных единиц. Это означает, что почтовой службой может быть зарегистрированное письмо, заказное письмо, письмо или посылка (соответствующего размера).

При моделировании предположите, что могут быть добавлены новые роли (например, Водитель от сторонних компаний) и другие почтовые службы (например, проверка возраста). Такие новые роли и другие почтовые службы еще не моделируются в этой задаче.

Вам разрешено использовать все классы из *инструкций по отправке* в этом упражнении.

Эта задача чувствительна к регистру. Это означает, что вывод чувствителен к регистру ввода.

Выведите все цены с округлением до двух знаков после запятой.

B.1 Интерактивный пользовательский интерфейс.

После запуска ваша программа берет на себя управление консолью с помощью Terminal.readLine (), которые более подробно описаны ниже. После обработки команды ваша программа ожидает дальнейших команд до тех пор, пока в какой-то момент программа не завершится вводом строки символов **quit**.

Предположите, что когда вы запускаете свою программу, ни пользователи, ни почта не регистрируются и не хранятся в системе почтового отделения.

Сообщения об ошибках

Убедитесь, что спецификации в задаче не нарушены, выполнив следующие команды, и в этих случаях выдайте содержательное сообщение об ошибке.

Сообщение об ошибке также должно выводиться, если вводимые пользователем данные не соответствуют указанному формату. После вывода сообщения об ошибке программа должна продолжить работу, как и ожидалось, и дождаться следующего ввода. Каждое сообщение об ошибке должно начинаться с **Error** и не

должно содержать разрывов строки. Вы можете выбрать другой текст сообщения об ошибке, но он должен иметь смысл.

Автоматические тесты

Поскольку мы проводим автоматические тесты вашего интерактивного пользовательского интерфейса, результат должен точно соответствовать спецификациям. В частности, буквы нижнего и верхнего регистра, а также пробелы и разрывы строк должны точно совпадать. Также не предоставляйте никакой дополнительной информации.

Примеры взаимодействий

Номера строк и разделительная линия не являются частью пользовательского интерфейса, они служат только в качестве руководства для данного примера взаимодействия. Строки ввода обозначаются знаком **>** (знак “больше”), за которым следует пробел; эти два символа также не являются частью введенной команды, а служат только для различения строк ввода и вывода. Обратите внимание, что примеры последовательностей отдельных команд необходимо просматривать независимо друг от друга.

Placeholder

Обратите внимание, что при описании ввода и вывода слова в квадратных скобках (<и>) обозначают заполнители (placeholder), которые заменяются значениями для конкретного ввода и вывода. Эти фактические значения не содержат угловых скобок для ввода и вывода. Также сравните соответствующие образцы процессов.

B.2 команды

B.2.1 команда add – customer.

Команда add - customer добавляет в систему нового клиента. Эта операция может быть выполнена только в том случае, если ни один пользователь (клиент, сотрудник почтового отделения или сотрудник колл-центра) не вошел в систему с помощью команды аутентификации **authenticate**.

Ввод / Input

add - customer <First name>; <Last name>; <User name>; <Password>;<Perso>

<First name> и <Last name> описывают клиента и представляют собой любую строку String без разрыва строки и без точки с запятой.

<Username> - это строка Srting без разрыва строки и без точки с запятой, содержащая от 4 до 9 цифр для уникальной идентификации клиента в системе (см. Уникальную идентификацию почтового работника и сотрудника колл-центра по <personal number> с помощью команды *add - mailman* и командой *add - agent*).

<password> - любая строка String без разрыва строки и без точки с запятой, содержащая от 4 до 9 цифр.

<Perso> представляет собой номер удостоверения личности и представляет собой строку String без разрыва строки и без точки с запятой с 9 цифрами для уникальной идентификации клиента, которая может содержать буквы и цифры. Имя пользователя и номер ID-карты должны отличаться по соображениям защиты данных.

Вывод / Output

Если новый клиент был успешно добавлен в систему, возвращается **OK**. В случае ошибки (например, если существует клиент с указанным номером удостоверения личности), выводится сообщение об ошибке, начинающееся с **Error**.

B.2.2 команда add - mailman

Команда *add - mailman* добавляет в систему нового почтальона. Эта операция может быть выполнена только в том случае, если ни один пользователь (клиент, сотрудник почтового отделения или сотрудник колл-центра) не вошел в систему с помощью команды *authenticate*.

Ввод / Input

add - mailman <First name>; <Last name>; <Personal number>; <Password>

< First name > и < Last name> описывают почтового работника и представляют собой любую строку String без разрыва строки и без точки с запятой.

< Personal number > - положительное целое число Integer для четкого представления почтового работника.

< Password > - любая строка String без разрыва строки и без точки с запятой, содержащая от 4 до 9 цифр.

Вывод / Output

Если новый почтовый работник был успешно добавлен в систему, выводится **OK**. В случае ошибки (например, если сотрудник с указанным персональным номером уже существует) выводится сообщение об ошибке, начинающееся с *Error*.

B.2.3 команда add - agent

Команда *add - agent* добавляет в систему нового агента центра обработки вызовов. Эта операция может быть выполнена только в том случае, если ни один пользователь (клиент, сотрудник почтового отделения или сотрудник колл-центра) не вошел в систему с помощью команды аутентификации authenticate.

Ввод / Input

add - agent <First name>; <Last name>; <Personnel number>; <Password>

< First name > и < Last name> описывают сотрудника центра обработки вызовов и представляют собой любую строку String без разрыва строки и без точки с запятой.

<Personal number > - положительное целое число Integer для четкого представления сотрудника центра обработки вызовов.

< Password > - любая строка String без разрыва строки и без точки с запятой, содержащая от 4 до 9 цифр.

Вывод / Output

OK возвращается при условии, что новый агент центра обработки вызовов был успешно добавлен в систему. В случае ошибки (например, если сотрудник с указанным персональным номером уже существует) выводится сообщение об ошибке, начинающееся с ***Error***.

B.2.4 команда authenticate

Команда authenticate аутентифицирует пользователя (клиента, почтового работника или сотрудника колл-центра) в системе почтового отделения и предоставляет ему определенные параметры для вызова команд в зависимости от сохраненных прав. Только один пользователь может быть зарегистрирован в системе почтового отделения одновременно. Прежде чем новый пользователь успешно аутентифицирует себя с помощью команды authenticate, ранее вошедший в систему пользователь должен выйти из системы с помощью команды logout. Пример: клиент A вошел в систему. Если клиент B или сотрудник хочет войти в систему, клиент B должен сначала выйти из системы с помощью команды logout.

Ввод / Input

authenticate <Username>; <Password>

<User name> - это строка String, представляющая личное имя пользователя (имя пользователя клиента или табельный номер сотрудника).

<Password>- строка String, представляющая личный пароль пользователя.

Вывод / Output

В случае успешной аутентификации зарегистрированного пользователя выводится OK.

В случае ошибки (например, если пользователь уже вошел в систему) выводится сообщение об ошибке, начинающееся с Error. Недопустимое имя пользователя и пароль также должны присутствовать в сообщении об ошибке, начинающемся с Error.

B.2.5 команда logout

Команда logout выводит из системы пользователя, который ранее был аутентифицирован с помощью команды authenticate, и лишает его возможности вызывать соответствующие команды до тех пор, пока он не будет повторно аутентифицирован.

Ввод / Input

logout

Вывод / Output

Если ранее успешно аутентифицированный пользователь вышел из системы, выводится *OK*.

В случае ошибки (например, если пользователь, который должен выйти из системы, не был заранее аутентифицирован), выводится сообщение об ошибке, начинающееся с *Error*.

B.2.6 send - mail command

Команда send-mail позволяет клиенту или почтовому работнику, действующему от имени клиента, отправить почтовую службу от отправителя к получателю.

Эта операция может быть выполнена только в том случае, если клиент (отправитель) или почтовый работник ранее вошел в систему с помощью команды аутентификации authenticate и еще не вышел из системы.

Если зарегистрированный клиент отправляет письмо другому зарегистрированному клиенту, формат ввода следующий:

Вход (заказчик)

send - mail <Postal service>; <Recipient>

<Postal service> - это строка String, состоящая из { Letter, registered mail, registered mail, PaketS, PaketM, PaketL } без разрыва строки и без точки с запятой.

< Recipient > - зарегистрированный клиент (см. Команду add - customer) в системе, который представлен в этой команде своим уникальным именем пользователя <User name >.

Если зарегистрированный почтовый служащий отправляет почту от имени клиента, формат ввода расширяется, чтобы включить связанный <Sender>:

Вход (работник почты)

send - mail <Postal service>; <Recipient>; <Sender>

<Sender> также является зарегистрированным клиентом (см. команда add-customer) в системе, который представлен в этой команде своим уникальным именем пользователя < user name >.

Вывод (клиент и почтовый работник)

В случае успеха выдается *OK*. В случае ошибки выводится сообщение об ошибке, начинающееся с *Error.*

B.2.7 команда get - mail

Команда get-mail позволяет клиенту и почтовому служащему, действующему от имени клиента, принимать почту, которая хранится в системе для соответствующего клиента <Recipient>, со станции в почтовом отделении. Эта операция может быть выполнена только в том случае, если клиент или почтовый работник ранее вошел в систему с помощью команды аутентификации authenticate и еще не вышел из системы.

Вход (заказчик)

get - mail

Вход (работник почты)

get – mail <Recipient>

Вывод (клиент и почтовый работник)

Предполагая, что почта, которая была депонирована для клиента, была успешно удалена с почтового отделения клиентом или почтовым работником, выдается *OK*. Если для зарегистрированного клиента не было сдано на хранение ни одной почты или если есть другие ошибки, выдается сообщение об ошибке, начинающееся с *Error*.

B.2.8 команда list-mail

Команда list-mail выводит отсортированный список почты, которая в настоящее время хранится в почтовом отделении для клиента и задокументирована в системе. Эта команда может выполняться клиентом, а также сотрудниками почты и колл-центра, которые действуют от имени клиента. Формат ввода для клиента следующий:

Input (customer) *list - mail*

Для зарегистрированного и вошедшего в систему сотрудника почты и колл-центра формат ввода следующий:

Input (post and call center employees/ сотрудники почты и колл-центра) *list - mail <User name>*

< User name > - это уникальное имя пользователя, уже зарегистрированного через add – customer.

Вывод (клиент, сотрудники почтового отделения и колл-центра)

<Postal service>; <Number>

Будет выведен список, отсортированный в алфавитном порядке в < Postal service>

< Postal Service > - это строка String, состоящая из { letter, registered mail, registered, parcelS, parcelM, parcelL} без разрыва строки и без точки с запятой.

<Number> описывает количество доступных почтовых услуг. Почтовые службы с номером 0 не выдаются.

Если нет сообщения для зарегистрированного клиента, выдается только ***OK***. В случае ошибки (например, клиент не существует) выводится сообщение об ошибке, начинающееся с ***Error***.

B.2.9 команда list-price

Команда list-price выводит совокупную цену, которую клиент уже заплатил за использование определенного типа услуги. Эта операция может быть выполнена, если клиент, имеющий имя пользователя < User name > в системе, ранее вошел в систему с помощью команды *authenticate* и еще не вышел из системы. Для клиента эта команда не имеет параметров.

Вход / Input (заказчик) *list – price*

Зарегистрированный сотрудник почты и колл-центра также может выполнить этот запрос от имени клиента. Для этого необходимо предоставить заказчику <user name >:

Вход / Input (сотрудники почты и колл-центра) *list - price <User name>*

Выход/Output (клиент, сотрудники почтового отделения и колл-центра)

<Postal service>; <Quantity>; <Price>

Если клиент уже отправил почтовые услуги через почтовое отделение, выводится список отправленных им писем, отсортированных в алфавитном порядке в соответствии с < *Postal service* >.

< Postal Service > - это строка , состоящая из {Letter, registered mail, ParcelS, ParcelM, ParcelL} без разрыва строки и без точки с запятой.

<Number> - положительное целое число Integer, начинающееся с 1, которое указывает количество отправленных писем для каждого типа службы.

<Price > - это совокупная цена, которую клиент уже заплатил за использование определенного типа почтовой услуги.

Если клиент еще не отправил почту (для данного типа услуги), дается только ***OK.*** В случае ошибки выводится сообщение об ошибке, начинающееся с ***Error***.

B.2.10 команда reset-pin

С помощью команды reset-pin сотрудник колл-центра может заменить пароль клиента новым паролем. Эта операция может быть выполнена только в том случае, если сотрудник центра обработки вызовов ранее входил в систему с помощью команды аутентификации и еще не вышел из системы.

Ввод :

reset - pin <user name>; <password>

< user name > - это имя пользователя, уже зарегистрированного добавлением клиента add-customer.

<password> - любая строка без разрыва строки и без точки с запятой, содержащая от 4 до 9 цифр.

Вывод:

Предполагая, что вошедший в систему сотрудник колл-центра заменил пароль для зарегистрированного клиента, выводится ***OK.***

Пример взаимодействия:

```
> reset-pin Erika;123
Error, incorrect input format, password is too short.
> reset-pin Erika;pass2
OK
```

Команда quit

Команда quit без параметров позволяет в любой момент полностью завершить программу. Обратите внимание, что для этого нельзя использовать такие методы, как System.exit () или Runtime.exit ()

Ввод / Input

Quit

Вывод / Output

В случае успеха нет вывода.

Пример взаимодействия:

```
> quit quit
Error, incorrect input format, this command does not accept any parameters.
> quit
```

B.3 Пример взаимодействия

Обратите внимание, что далее строки ввода обозначаются знаком **>**, за которым следует пробел. Эти два символа явно не являются частью введенной команды, а служат только для различения ввода и вывода.

```
> add-customer Max;Mustermann;MaxM;pass1;123456789
OK
> add-customer Erika;Mustermann;Erika;pass2;234567890
OK
> add-mailman Heinrich;Hinz;1;pass1
OK
> add-agent Konrad;Kunz;2;pass2
OK
> authenticate Erika;pass2
OK
> send-mail Brief;MaxM
OK
> list-price
Brief;1;0.70
> logout
OK
> quit
```