1. Прочитать

Исключения в Java, Часть I и II

<https://habrahabr.ru/company/golovachcourses/blog/223821/>

<https://habrahabr.ru/company/golovachcourses/blog/225585/>

https://dev64.wordpress.com/2012/10/07/exception-handling-antipatterns/

1. Реализовать интерфейс Terminal, c помощью которого можно:

* Проверить состояние счета
* Снять/ положить деньги

Доступ к терминалу (терминал для одного счета) предоставляется только после ввода корректного пин-кода (4 цифры). Каждое нажатие должно восприниматься системой как ввод очередного символа пин-кода.  
При вводе нецифрового символа система должна выдать предупреждение, но при этом данное действие не считается критичной ошибкой ввода и состояние системы должно восстановиться к последнему корректному состоянию.

При вводе 3 неправильных пин-кодов аккаунт блокируется на 10 сек (при попытке обратиться к нему в течение этого времени должно вызываться исключение AccountIsLockedException c информацией об оставшемся времени до снятия блокировки).

Класть и снимать деньги можно только, если сумма кратна 100.

Поскольку банкоматы, установленные на улице, сами «ничего не делают» с вашим счетом (они лишь делают проверку введенных данных и отправляют запросы на удаленный), то сервер (TerminalServer) может вызывать свои исключения, например, при недостатке средств на счете для совершения операции.

класс терминал может содержать следующие поля:

class TerminalImpl {

    private final TerminalServer server;

    private final PinValidator pinValidator;

   ...

}

Часть команд терминал делегирует этим классам.

Интерфейс терминала и список исключений остается на ваш дизайн. В каждом ексепшене должно быть описание, что нужно сделать, чтобы избежать его в дальнейшем.

TerminalServer и PinValidator могут кидать свои собственные исключения. Конечный пользователь не должен видеть эти исключения, ему должны показываться человеко-читабельные сообщения об ошибках

Логика по показу сообщений конечному пользователю и бизнес логика терминала с обработкой исключений должна быть в разных классах, чтобы можно было легко менять интрефейс вывода сообщений (на консоль, через графический интерфейс и т.д.)

1. Реализуйте метод readContent(String url), который отображает на экран

содержимое сайта, ссылка на который задаётся параметром url.

Напишите программу, считывающую из консоли строку (URL ресурса) и вызывающую

метод readContent. В том случае, если введённый URL неправильного формата

или нет доступа до указанного ресурса, пользователю предлагается повторить ввод.

1. Вопросы для самоконтроля:
2. Правильно ли написан следующий код:

try {

} finally {

}

2. Какие исключения могут быть перехвачены следующим обработчиком:

catch (Exception e) {

}

Что не так с использованием такого обработчика?

3. Имеется ли ошибка в написании такого обработчика исключения? Скомпилируется ли этот код?

try {

} catch (Exception e) {

} catch (ArithmeticException a) {

}

4.Сопоставьте каждую ситуацию из первого списка с пунктом второго списка:

а) int[] A; A[0] = 0;

б) JVM начинает запускать вашу программу, но обнаруживает, что не может найти классы Java платформы;

в) программа читает входной поток и достигает маркера конца потока;

г) после достижения маркера конца входного потока, но до закрытия самого потока, программа пытаться считать данные.

1) ошибка;

2) проверяемое исключение;

3) ошибка компиляции;

4) исключение не произойдёт.