***1. Какие данные могут хранить коллекции?***

Коллекциями/контейнерами в Java принято называть классы, основная цель которых – хранить набор других элементов.

Коллекции могут хранить любые ссылочные типы данных.

Каждый класс это свой тип данных. Ссылочный тип данных.  
A a = new A(); // Тепрь переменная «a» — указывает на новый объект, тип которого A. На самом деле «a» хранит адрес в памяти, где хранится новый объект. Но от программиста адрес «скрыт». Всю работу с адресами делает java.

***2. Как написать собственное (“пользовательское”) исключение? Какими мотивами вы будете руководствоваться при выборе типа исключения: checked/unchecked?***

Необходимо унаследоваться от базового класса требуемого типа исключений (например от Exception или RuntimeException).

public class ExcClass extends Exception {

    private String someString;

    public ExcClass (String string) {

        this.someString = string;

        System.out.println("Exception ExcClass");

    }

    public void myOwnExceptionMsg() {

        System.err.println("This is exception message for string: " + someString);

    }

}

public class TestExc {

    public static void main(String[] args) {

        try {

            String s = "SomeString";

            throw new ExcClass(s);

        } catch (ExcClass ex) {

            ex.myOwnExceptionMsg();

        }

    }

}

//Вывод

Exception ExcClass

This is exception message for string: SomeString

Руководствоваться нужно определением типа исключения. В зависимости от того, что вы хотите обрабатывать или видеть нужно и наследоваться от нужного класса.

Механизм исключительных ситуация в Java связан с двумя элементами «магии», т.е. поведения, которое никак не отражено в исходном коде:  
1. «Магию» java.lang.Throwable — в throw, catch и throws могут стоять исключительно Throwable или его наследники ([мы уже разбирали в предыдущей лекции](http://habrahabr.ru/company/golovachcourses/blog/223821/#1)). Это «право» находиться в throw, catch и throws никак не отражено в исходном коде.  
2. Все исключительные ситуации делятся на «проверяемые» (checked) и «непроверяемые» (unchecked). Это свойство присуще «корневищу» (Throwable, Error, Exception, RuntimeException) и передается по наследству. Никак не видимо в исходном коде класса исключения.

Расставим значение свойства checked/unchecked