Исключения. Работа с датой

Цели: ознакомиться с коллекциями в Java

План урока:

- статические члены класса
- работа с датой
- понятие стека вызовов
- исключения

Static

Ключевым словом static помечают члены (поля или методы), которые **принадлежат классу, а не экземпляру этого класса.**

- Это означает, что какое бы количество объектов вы не создали, всегда будет создан только один член, доступный для использования всеми экземплярами класса.
- Ключевое слово static применимо к переменным, методам, блокам инициализации, импорту и вложенным классам (nested classes).

В языке Java, если поле объявляется статическим (путем добавления модификатора static), то в независимости от количества созданных объектов класса — всегда будет существовать только один экземпляр статического поля. **Значение такого поля будет единым и общим для всех объектов класса, содержащих это поле.**

Подобно *статическим* полям, ***статические*** **методы** также принадлежат классу, а не объекту, поэтому их можно вызывать без создания экземпляра класса, в котором они находятся. При этом следует помнить, что из статического метода можно получить доступ только к статическим переменным или к другим статическим методам

Exceptions

Исключения (исключительные ситуации) - ошибки, возникшие в программе во время её работы.

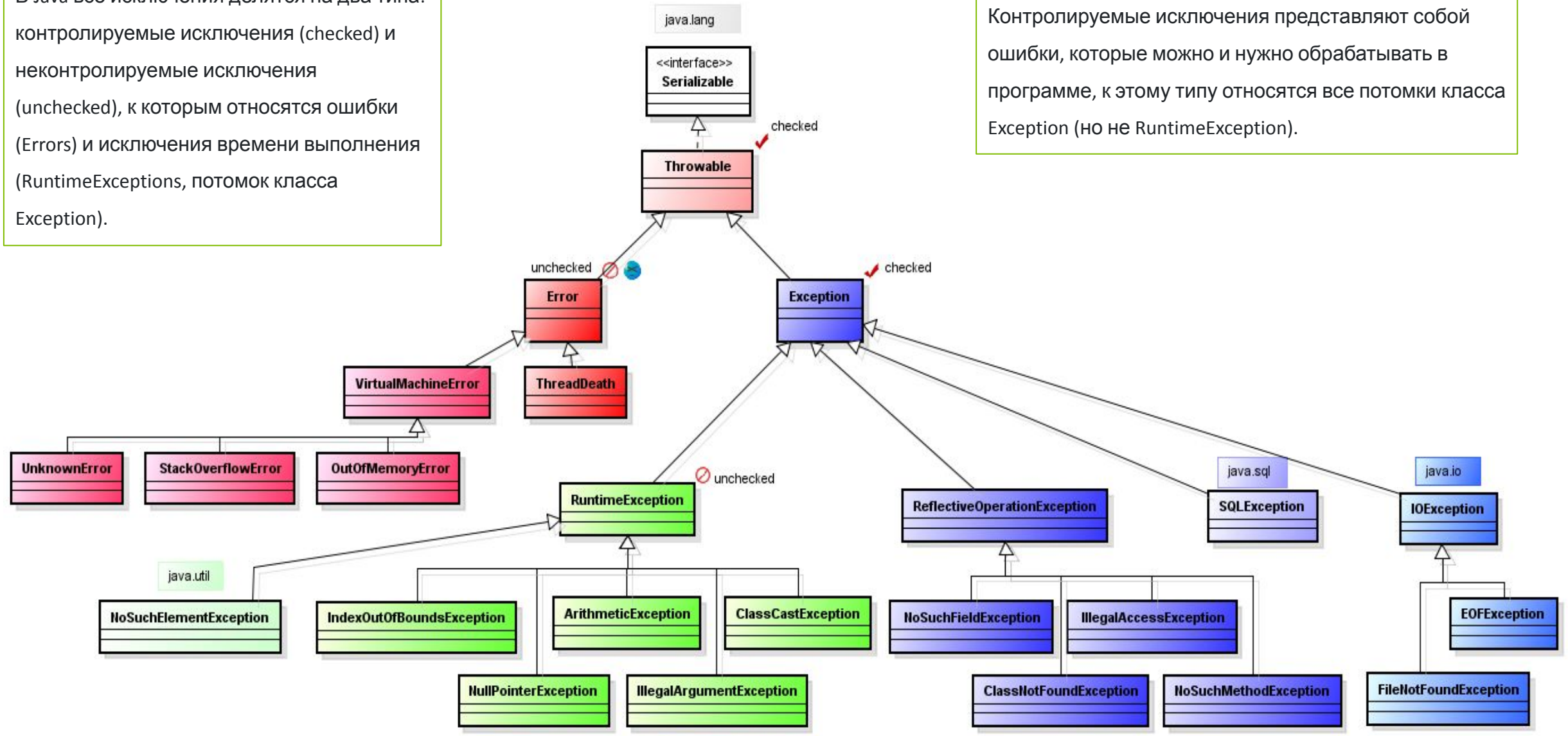
Все исключения в Java являются объектами. Поэтому они могут порождаться не только автоматически при возникновении исключительной ситуации, но и создаваться самим разработчиком.

Исключения (Exceptions) являются результатом проблем в программе, которые в принципе решаемы и предсказуемы. Например, произошло деление на ноль в целых числах.

Ошибки (Errors) представляют собой более серьёзные проблемы, которые, согласно спецификации Java, не следует пытаться обрабатывать в собственной программе, поскольку они связаны с проблемами уровня JVM. Например, исключения такого рода возникают, если закончилась память, доступная виртуальной машине. Программа дополнительную память всё равно не сможет обеспечить для JVM.

В Java все исключения делятся на два типа: контролируемые исключения (checked) и неконтролируемые исключения (unchecked), к которым относятся ошибки (Errors) и исключения времени выполнения (RuntimeExceptions, потомок класса Exception).

Контролируемые исключения представляют собой ошибки, которые можно и нужно обрабатывать в программе, к этому типу относятся все потомки класса Exception (но не RuntimeException).



Операторы для работы с исключениями

Обработка исключения может быть произведена с помощью операторов **try…catch**, либо передана внешней части программы. Например, метод может передавать возникшие в нём исключения выше по иерархии вызовов, сам его не обрабатывая.

В Java есть пять ключевых слов для работы с исключениями:

1. **try** — данное ключевое слово используется для отметки начала блока кода, который потенциально может привести к ошибке.
2. **catch** — ключевое слово для отметки начала блока кода, предназначенного для перехвата и обработки исключений.
3. **finally** — ключевое слово для отметки начала блока кода, которой является дополнительным. Этот блок помещается после последнего блока ‘catch’. Управление обычно передаётся в блок ‘finally’ в любом случае.
4. **throw** — служит для генерации исключений.
5. **throws** — ключевое слово, которое прописывается в сигнатуре метода, и обозначающее что метод потенциально может выбросить исключение с указанным типом.

Синтаксис

```
try{
//здесь код, который потенциально может привести к ошибке
}
catch(SomeException e ){ //в скобках указывается класс конкретной ожидаемой ошибки
//здесь описываются действия, направленные на обработку исключений
}
finally{
//выполняется в любом случае ( блок finally  не обязателен)
}
```

```
public static int getFactorial(int num)  throws Exception{
    try {
        // ...
        throw new ThrowableClass(parameters); // если исключительная ситуация, то сгенерировать
    исключение
        // ...
    }
    catch (ThrowableClass e) {
        // обработать исключение
        // ...
    }
}
```

Вопросы?