СБЕРБАНК ТЕХНОЛОГИИ

Classloaders

УЗНАЕМ

СБЕРБАНК ТЕХНОЛОГИИ

- Когда и как загружаются Java классы?
- Какие есть стандартные классслоадеры?
- Ошибки при работе с загрузкой классов

КОГДА JVM ЗАГРУЖАЕТ КЛАССЫ ?

СБЕРБАНК ТЕХНОЛОГИИ

JVM загружает класс при первом обращении к нему

(создание объекта класса, обращение к статик методу/полю…)

ГДЕ ХРАНЯТСЯ КЛАССЫ?

СБЕРБАНК ТЕХНОЛОГИИ

Загруженный класс хранится в области Permgen.
В Java 8 в Metaspace.

https://habrahabr.ru/post/169869/ Java 8: от PermGen к MetaSpace

ЧТО ТАКОЕ CLASSLOADER

СБЕРБАНК ТЕХНОЛОГИИ

Класслоадеры - это классы, экземпляры которых, загружают другие классы.

Отличаются логикой загрузки(Из БД, файла, по сети..)

СТАНДАРТНЫЕ КЛАССЛОАДЕРЫ

СБЕРБАНК ТЕХНОЛОГИИ

- Bootstrap
- Extensions
- System

BOOTSTRAP CLASSLOADER

СБЕРБАНК ТЕХНОЛОГИИ

Загружает классы JDK (String, Integer, List…).

Является частью JVM.

Единственный класслоадер, реализованный не на Java (Си).

EXTENSIONS CLASSLOADER

СБЕРБАНК ТЕХНОЛОГИИ

Загружает классы из папки <JAVA_HOME>/jre/lib/ext

(Папки можно задавать через параметр -Djava.ext.dirs)

Реализован в sun.misc.Launcher$ExtClassLoader.

Почти не используется.

SYSTEM/APPLICATION CLASSLOADER

СБЕРБАНК ТЕХНОЛОГИИ

Загружает классы из classpath.

Реализован в sun.misc.Launcher$AppClassLoader.

Полезный java.net.URLClassLoader

СБЕРБАНК ТЕХНОЛОГИИ

Загружает файлы по сети или с файловой системы

```
//Пути до классов передаются в конструкторе
public URLClassLoader(URL[] urls) {…}
```

java.lang.Classloader основные методы

СБЕРБАНК ТЕХНОЛОГИИ

```
public abstract class ClassLoader {
    public Class<?> loadClass(String fullClassName)
                throws ClassNotFoundException {…}


    protected Class<?> defineClass(byte[] b, …)


    public ClassLoader getParent()
}
```

Какой класслоадер начинает загрузку класса?

СБЕРБАНК ТЕХНОЛОГИИ

```
public class Main {
    public void run {
        new Person();

        //Равносильно
        Main.class.getClassLoader().loadClass("Person")
    }
}
```

У класслоадера есть родительский класслоадер.

У System это Extensions.
У Extensions это Bootstrap.
У Bootstrap нет родителя.

Модель делегирования

СБЕРБАНК ТЕХНОЛОГИИ

System classloader

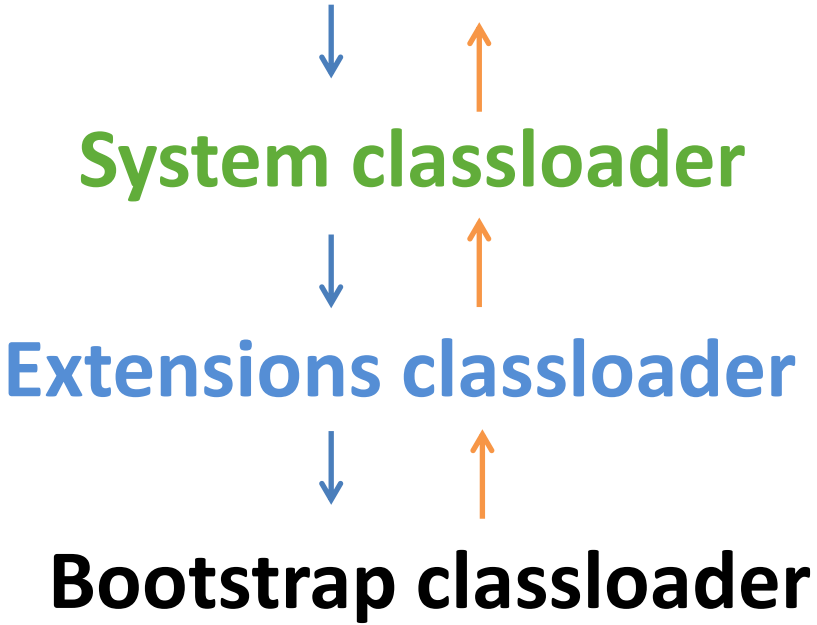
Extensions classloader

Bootstrap classloader

CL проверяет, загружал ли он этот класс.
Если загружал, возвращает его.

Иначе делегирует своему родителю.

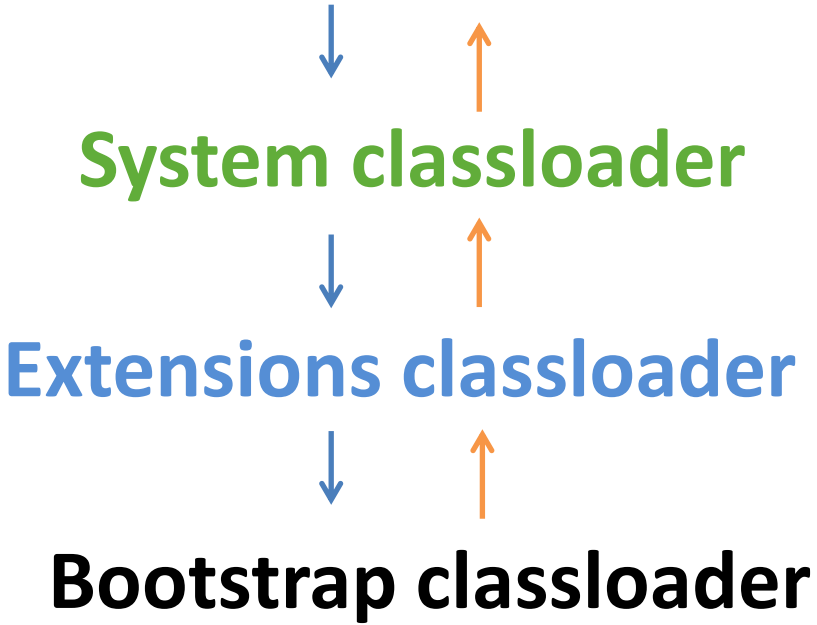
Модель делегирования



CL пытается сам загрузить класс.

Если не получается, делегирует назад своему потомку

Модель делегирования



Если класс не смог быть загружен,
кидается ClassNotFoundException

Classloader::loadClass реализация

СБЕРБАНК

```
protected Class<?> loadClass(String name)
                              throws ClassNotFoundException {
    Class<?> c = findLoadedClass(name);
    if (c == null) {
        try {
            if (parent != null) {
                c = parent.loadClass(name, false);
            } else {
                c = findBootstrapClassOrNull(name);
            }
        } catch (ClassNotFoundException ignore) {
        if (c == null) c = findClass(name);
    }
    return c;
}
```

МОДЕЛЬ ДЕЛЕГИРОВАНИЯ ЕЩЕ РАЗ

СБЕРБАНК ТЕХНОЛОГИИ

При загрузке класса классloадер:

1. Проверяет, загружал ли он этот класс. Если загружал, возвращает его.
2. Иначе просит своего родителя загрузить класс.
3. Шаги 1 и 2 повторяются, то тех пор пока цепочка вызовов не дошла до последнего классloадера.
4. Родительский классloадер пытается загрузить класс, если не получается - возвращает управление своему Child’у.
5. Если класс не смог быть загружен, кидается ClassNotFoundException.

INITIATING И DEFINING CLASSLOADERS

СБЕРБАНК ТЕХНОЛОГИИ

Initiating - Класслоадер, который начал загрузку.

Defining – Класслоадер, который реально загрузил класс (после делегации).

УНИКАЛЬНОСТЬ КЛАССА

СБЕРБАНК ТЕХНОЛОГИИ

Уникальность класса определяется полным именем и defining классслоадером.

КОГДА КЛАСС МОЖЕТ БЫТЬ ВЫГРУЖЕН?

СБЕРБАНК ТЕХНОЛОГИИ

- Когда нет доступных ссылок на класс, объектов этого класса.
- Нет ссылок на классслоадер, загрузивший этот класс.

Person.class МОЖЕТ БЫТЬ СРАЗУ ВЫГРУЖЕН

СБЕРБАНК ТЕХНОЛОГИИ

```
public void run() throws ClassNotFoundException {
    for (int i = 0; i < 1_000_000; i++) {
        new SomeClassLoader().loadClass("ru.sbt.Person");
    }
}
```

ClassNotFoundException

СБЕРБАНК ТЕХНОЛОГИИ

- ClassNotFoundException – класс не найден при попытки загрузить его вручную по имени

```
ClassLoader.getSystemClassLoader()
                    .loadClass("ru.sbt.school.Main")
```

NoClassDefFoundError

СБЕРБАНК ТЕХНОЛОГИИ

- NoClassDefFoundError – класс был доступен на этапе компиляции, в рантайме не был найден

```
public static void main(String[] args) {
    Person person = new Person(); //Возможен NCDFE
}
```

Разные версии класса

СБЕРБАНК ТЕХНОЛОГИИ

- NoSuchMethodError – На этапе компиляции и в рантайме присутствовали разные версии класс
- IllegalAccessError – Загружена другая версия класса, в котором у метода изменили область видимости

```
Person person = new Person();
person.sayHello(); // Код был скомпилирован,
                   // у загруженной версии класса
                   // данного метода нет
                   // или он стал private/protected
```

ClassCastException

СБЕРБАНК ТЕХНОЛОГИИ

- ClassCastException – попытка присвоения, когда классы загружены разными классloадерами

```
Person person = (Person) new SomeClassLoader()
            .loadClass("ru.sbt.Person").newInstance();
```

Exception in thread "main" java.lang.ClassCastException:
ru.sbt.Person cannot be cast to ru.sbt.Person

ЗАГРУЗКА ИЕРАРХИИ КЛАССОВ

СБЕРБАНК ТЕХНОЛОГИИ

При запросе на загрузку класса неявно начинается загрузка всех его суперклассов и интерфейсов.

Суперклассы могут быть загружены **родительскими классloадерами.**

ЗАГРУЗКА ИЕРАРХИИ КЛАССОВ

СБЕРБАНК ТЕХНОЛОГИИ

```
public class CalculatorImpl implements Calculator
```

При загрузке `CalculatorImpl` классслоадер сначала должен загрузить `Calculator`, а для этого сначала надо загрузить Object.

ДОПУСТИМ У НАС ЕСТЬ

СБЕРБАНК ТЕХНОЛОГИИ

загружен **System CL**

CalculatorImpl implements

загружен **System CL**

Calculator

ДОПУСТИМ У НАС ЕСТЬ

СБЕРБАНК ТЕХНОЛОГИИ

загружен **System CL**

CalculatorImpl implements Calculator

загружен **System CL**

CalculatorImpl implements Calculator

загружен **URL CL**

CalculatorImpl implements Calculator

загружен **URL CL**

CalculatorImpl implements Calculator

ДОПУСТИМ У НАС ЕСТЬ

СБЕРБАНК ТЕХНОЛОГИИ

загружен **System CL**
CalculatorImpl implements Calculator

загружен **System CL**
CalculatorImpl implements Calculator

загружен **URL CL**
CalculatorImpl implements Calculator

загружен **URL CL**
CalculatorImpl implements Calculator

загружен **URL CL**
CalculatorImpl implements Calculator

загружен **System CL**
CalculatorImpl implements Calculator

ЗАГРУЗКА ИЕРАРХИИ КЛАССОВ. CLASS CAST EXCEPTION

СБЕРБАНК ТЕХНОЛОГИИ

загружен **System CL** загружен **System CL**

```
CalculatorImpl implements Calculator
```

загружен **URL CL** загружен **URL CL**

```
CalculatorImpl implements Calculator
//Объект CalculatorImpl не может быть присвоен в
CalculatorImpl(System) или Calculator(System)
```

ЗАГРУЗКА ИЕРАРХИИ КЛАССОВ. CLASS CAST EXCEPTION

СБЕРБАНК ТЕХНОЛОГИИ

загружен **System CL** загружен **System CL**

```
CalculatorImpl implements Calculator
```

загружен **URL CL** загружен **URL CL**

```
CalculatorImpl implements Calculator
//Объект CalculatorImpl не может быть присвоен в
CalculatorImpl(System) или Calculator(System)
```

загружен **URL CL** загружен **System CL**

```
CalculatorImpl implements Calculator
//Объект CalculatorImpl не может быть присвоен в
CalculatorImpl(System), но может в Calculator(System)
```

ClassCastException пример

СБЕРБАНК ТЕХНОЛОГИИ

```
public void calc() {
    //Ok. Оба объекта загружены System CL
    doCalc(new CalculatorImpl());









}

private void doCalc(CalculatorImpl calculator) {
    calculator.calc();
}
```

ClassCastException пример

СБЕРБАНК ТЕХНОЛОГИИ

```
public void calc() {
    //Ok. Оба объекта загружены System CL
    doCalc(new CalculatorImpl());



    doCalc((CalculatorImpl) new CalcClassloader()
                                .loadClass("CalculatorImpl")
                                .newInstance());
}

private void doCalc(CalculatorImpl calculator) {
    calculator.calc();
}
```

ClassCastException пример

СБЕРБАНК ТЕХНОЛОГИИ

```
public void calc() {
    //Ok. Оба объекта загружены System CL
    doCalc(new CalculatorImpl());

    //ClassCastExcp. Присваивание класса, загруженного
    CalcClassLoader, в переменную загруженную System CL
    doCalc((CalculatorImpl) new CalcClassloader()
                                .loadClass("CalculatorImpl")
                                .newInstance());
}

private void doCalc(CalculatorImpl calculator) {
    calculator.calc();
}
```

Лечим

СБЕРБАНК ТЕХНОЛОГИИ

```
public void calc() {
    //Ok. Оба объекта загружены System CL
    doCalc(new CalculatorImpl());

    //Ok. Если при загрузке CalculatorImpl,
      загрузим Calculator с помощью System CL
    doCalc((Calculator) new CalcClassloader()
                              .loadClass("CalculatorImpl")
                              .newInstance());
}

private void doCalc(Calculator calculator) {
    calculator.calc();
}
```

МАТЕРИАЛЫ

СБЕРБАНК ТЕХНОЛОГИИ

https://vcfvct.files.wordpress.com/2015/06/do-you-really-get-classloaders.pdf