17.10.18

Самостоятельно пройденные темы повторить.

Чекнуть Boxing/Unboxing;

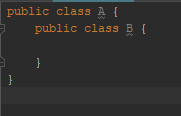
1. ООП:

Основные понятия

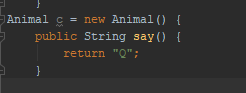
Классы объявляют новый ссылочный тип и определяют его реализацию.

Виды: Вложенный(member class – внутри класса, local class – внутри метода, anonymous class – без имени), верхнего уровня(не вложенный), именованный.

Вложенный:



Анонимный:



Любой класс – наследник Обджект. Содержит поля и методы. Часто встречаются анонимные классы.

Видимость регулируется модификаторами доступа. Public, private, package, protected;

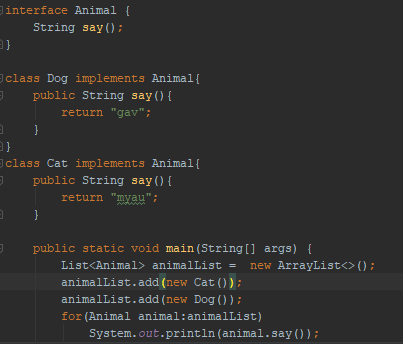
Public – доступ отовсюду.

Private – доступ только из текущего класса.

Package – в рамках пакета.

Protected – в пакете и классы-наследники.

Интерфейсы:



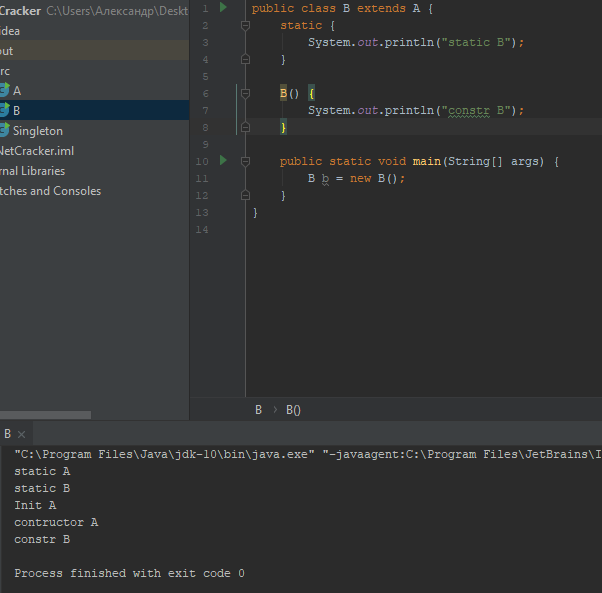
* Нет множественного наследования.
* Члены класса – унаследованные и определенные в классе члены
* Новые поля могут скрывать поля суперклассов и суперинтерфейсов
* Новые методы могут скрывать, реализовывать или перегружать методы суперкласса
* Вложенные классы мб статичскими и внутренними

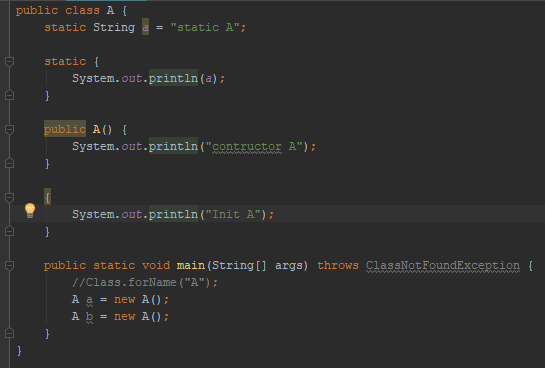
Полиморфизм – вариация поведения в зависимости от конкретной реализации в рамках единого контракта.

Private,static final – не полиморфны.

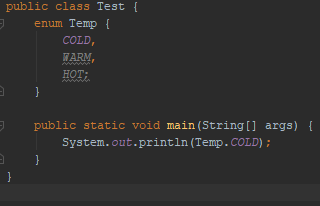
24.10.18.

Загрузка классов:

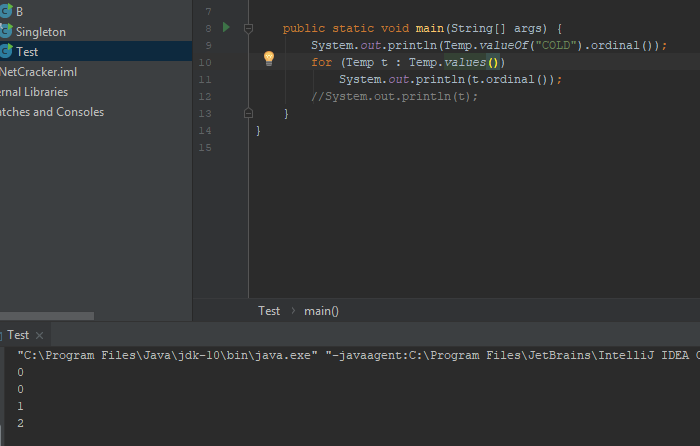




* Enum - нельзя создать экземпляр, но в java выглядешь будет как класс.



.values() – возвращает значения; .ordinal() – выводит порядковый номер элемента;



При компиляции : создается класс final abstract Temp extends Enum;

Java делится на:

1)SE – основы языка

2)EE – сервлеты, апплеты, jsp, JPA, …. Для коммерческих приложений. WEBLOGIC, Glassfish, Jboss – устаревает.

3)ME – девайсы.

Компиляция программы java:  
код – compile – байт код – JVM ;

Test.java -> Test.class;

Часто исполняемы код jvm может скомпилировать : JIT(just in time) .

**Class loader** идет в class path берет там классы.

Garbage Collector

**STRING**

String, StringBuffer, StringBuilder;

String – неизменяемый.

String - threadSafe

**StringBuilder- изменяемы.   
StringBuffer – mutable thread-safe;**

**StringTokenizer vs .split():**

**1 – быстрый , парсит данные, устаревший.**

**2 – меддерживает RegEx, возвращает массив, но медленнее**

**Pattern – позвояет искать взхождения строк/подстрок по шаблонам. Matcher;**

**String + RegEX check?>>??**

**ИСКЛЮЧЕНИЯ**

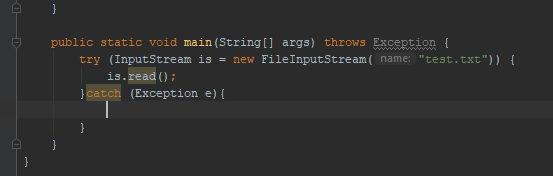
**Throw new IllegalArgumentException(); - выдаем исключение.**

**Исключения - тоже объекты.**

**Object<- [Throwable]<-Error –кидает сама JVM(StackOverFlow,OutOfMemory)**

**[Throwable]<-[Exception] – проверяемы и надо обрабатывать🡨[RuntimeException] – не checked**

**Try с ресурсами: try(InputStream is = new InputStream(“”)));**



**Посмотреть лекцию : bit.ly/2DhkujK**

**02.11.18**

Kafka, elastic search, open shift, hybern,mongo,postgree.

**Object**: wait, notify,notifyall, toString, hashcode,equals,clone,getClass,finalize,instance of;

**Wait – для многопоточки.Ожидает некоторого условия, Снимает блокировку с объекта.**

**Notify – снимает блокировку, выходит из wait .**

**Если переопределили equals =>Override hashcode.**

**GB:**

**3 поколения yoiunt old permanent;**

**B permanent – class, ??константы??? – НЕ ОЧИЩАЕТСЯ**

**YounG состоит из Eden, Ser1,Ser2(like from-to- space);**

**Minor GB – только с янг.**

**Major GC – old**

**Работа Minor GC:**

**1)Pause**

**2)Живые из eden -> to space**

**3)Из from spase ->to space/old Gen;**

**4) Clear eden and FROM space;**

**5)From space swapt TO space;**

**6)Continue;**

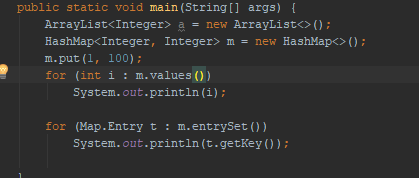
**Major:**

1. **Идем по дереву и помечаем живые**
2. **Мусорные в конец блока old Gen.**
3. **Очистить все после последнего живого.**

**Collections:**

**Map не от itarable, поэтому нельзя итерироваться.**

**Поэтому берем по ключам entrySet, or values() – возвращает collection, keySet() – возвращает Set;**



**Set:**

**HashSet,TreeSet, LinkedHashSet,CopyOnWriteArraySet;**

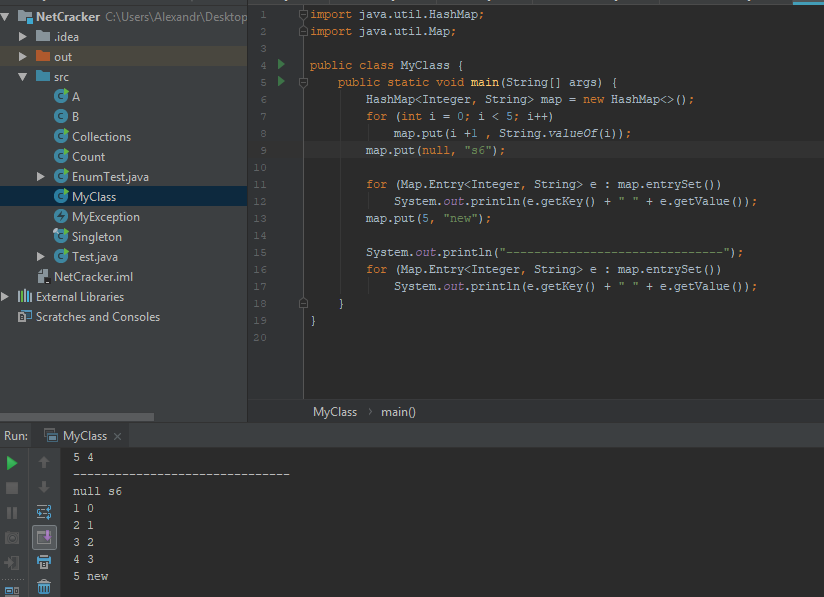
**List:**

**Можно добавлять в конретное место.**

**Объяснить причины**

**MAP**

**Hashmap:  
Бакеты, 16 штук по умолчанию. Определение корзины: считаем хэшкод ключа % количество корзин. Получили номер бакета. Сраниваем ключи. Кладем далее.**



**Итераторы**

**3 метода: hasnext,next,**

**Компаратор**

**Generics**

ДЗ: 1) Построить иерархию с эплом. Fruit->Apple->GrannyApple

->GreenApple

->Orange;

List<?extends Apple> что можем положить.

List<?super Apple> что можем положить.

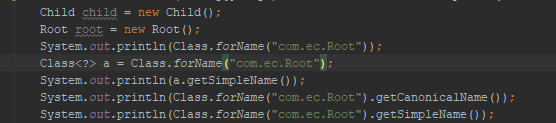
Тестирование

Selenium, Junit;

Test driven Development;

Mockito,Jenkins

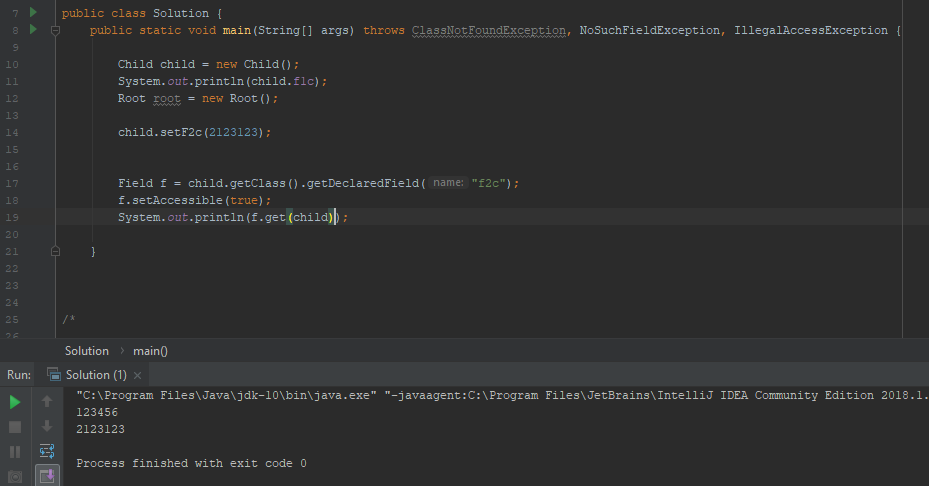
**Рефлексия**

Позволяет вытащить инфу о классе + методы и т.п



getMethod() – паблик методы + наследование.

getDeclaredMethod() – все методы без наследования



**Аннотации**

SOURCE CLASS RUNTIME

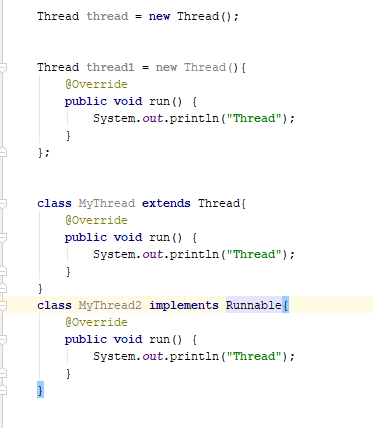
**SECURITY**

Opensource project OWASP

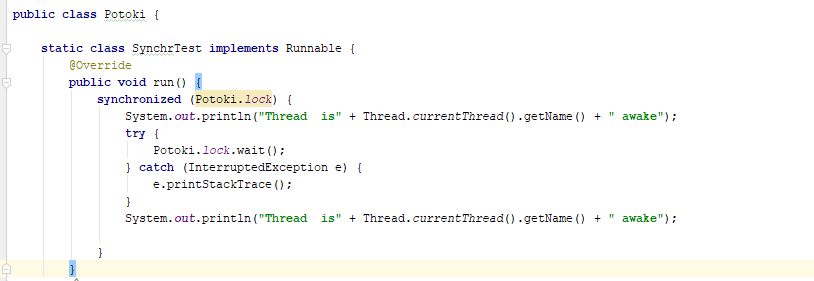
**Многопоточность**

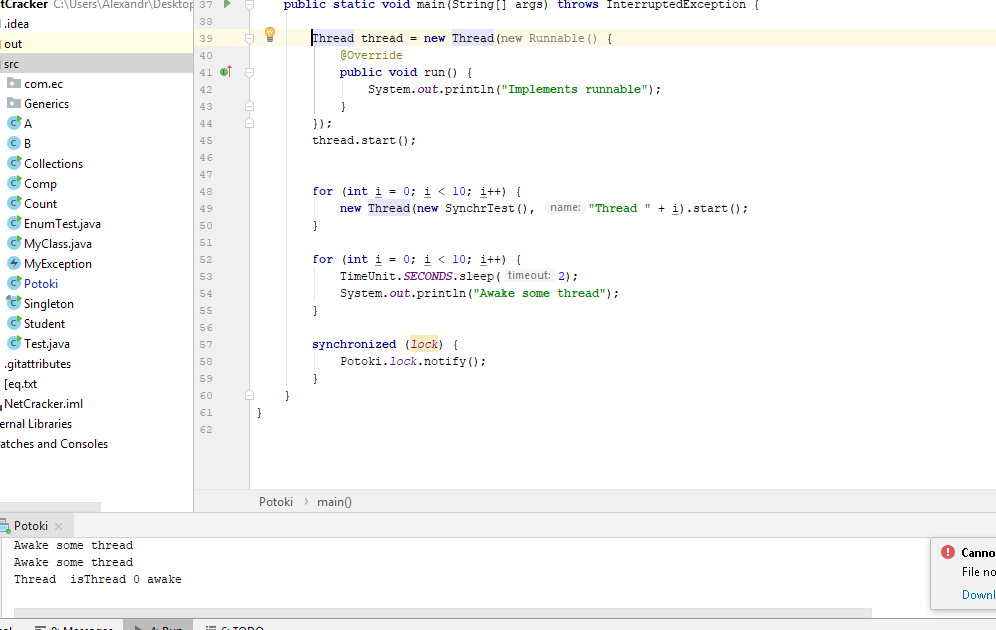
**Synchronized**

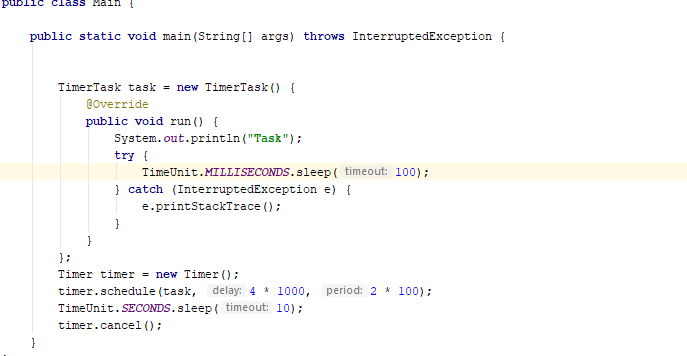
**Можно обозначить метод, можно кусок кода**



**.RUN – не создает новый поток, .START создает.**

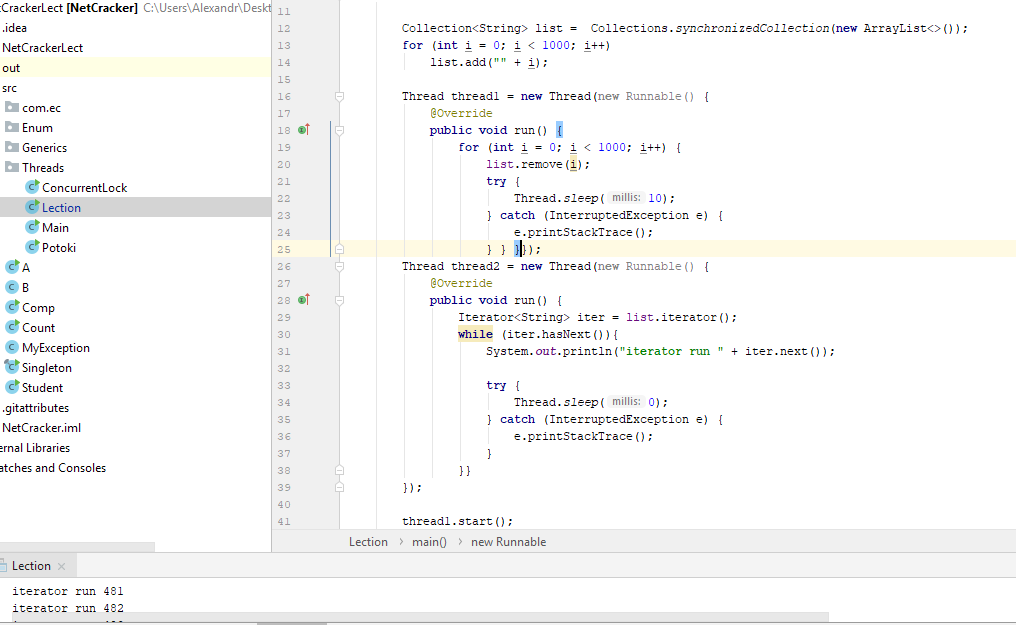






**Exchanger**

**Collections.syncronized:**



**Executors:**

**CachedTHreadPool**

**FixedThreadPool**

**SingleThreadPool**

**Compare and Swap**

**Cas(previous, new Val){**

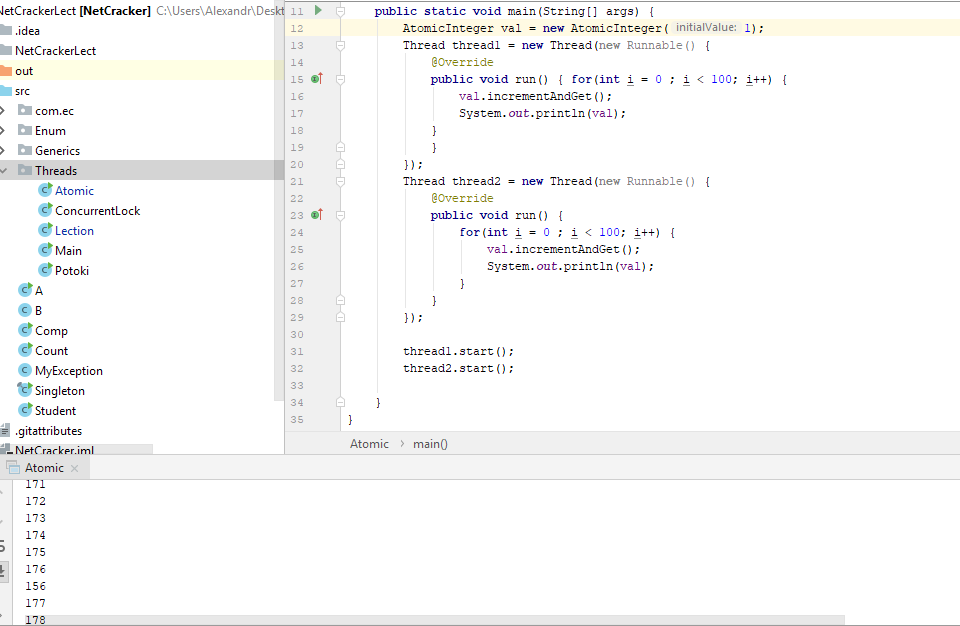
**while (current!=prev){**

**prev =current}**

**current = new**

**}**

**Это реализованно в Atomic**



**Collections в многопоточности**

**ConcurrentMap:**

**putIfAbsent; replace;remove;**

**CopyONWrite снованы на cas;**

**Input/Output Stream,Serialization,Files;**



**Serialization**

1. **Serializable – интерфейс. Реализация методов writeObject, readObject;**
2. **Интерфейс Externalizable**

**ДЗ: на флешке.**