pub. Cl A{}->A.java compiler(javac)

[Load, Store]->A.class İnterpreter(JVM) to code Justintime Compilation +Hotspot->impr. Perf. **+Operator**->only overloaded.

No argc, args.length

A a, a reference, a= new A(), A a = new A(), in String Str s = new String(“a”) **dont use==**str comp **StrBuffer** Mut, sync (mult.thr), **StrBuilder** mut, not sync **Array,** each dimension special obj. Size of arr cant modified. Intarr init as 0, others null, Nullpointerexc. dont use.

Circle[] ci = new C[size] foreach for init E:For(i..)ci[i] = newC(..) OK

E:For(i)ci[i].setRa(..)no, null no need

E:For(c:ci)c=newC() NO, to set

İnt[x][y] irr={{..},{..}}, length array(x,y)

**İmport static ...Math;** d=cos(), instd.of Math.cos() **Autoboxing**,has methods prim(double)->Cls(Double) **ReadInp**:ComLine(args), JoptionPane ,Scanner. In a file, only 1 pub.cl **Inheritance** Pub cl Child ext Par{}, child access **all non private** of Par, access private with get/set. If no extends-> extends Object. Construc: first line is **super()**, by default. @**Override**, comp detect. o.w. run time detection **Attempt1:**

sumAr(Obj[] ob){ for(..)sum+= ob.ar()} Fails. Ob don’t have getArea in its view. A,f()<-B,g(). B inherits A(Circ. İnh. Obj). When A a = new B(), it’s from A’s view.Can say a.f(), cant say a.g(). No g in A. **Attempt2:** Create Shape, getArea()=-1. Cir.ext Sha{..} Sq. Ext Sha{..}. Works

sumAr(Sha[]s]{ for(.)sum+= s.ar()}

Efficiency issue: shape is created.

**Attempt3: Abstract Class** make Sh. abs. Pub. Abs. Sh{}. Cant instantiate Sh. (no new Shape()). Can hava abs cl. w.o. abs. method. Cir ext Sh{ar()}, Sq ext Sh{ar()} **Attempt4: Interface** In ex. We don’t modify a data member. In java8, interfaces can have concrete(default) and static methods. No modifiebly attr. In ex: we can use Int. Bc we don’t modify a data member. Pub int. Sh{getA();

sta. sumAr(..){imp.});}

Abs.cl = Part of design, for me.

Inter: Public. All methods public by default.Pub int Sh{getA();}, if a cl. imp. Sh but not getA(), that class is abstract. Cl A imp IA{...}. IA ia = new A(); IA,f()<-A,g(). Same issue. Cant say ia.g(). g() does not exist in IA.

A cl ext 1 cl, imp many inter. Cl A imp IA1,IA2 -> multibehaviour inherit.

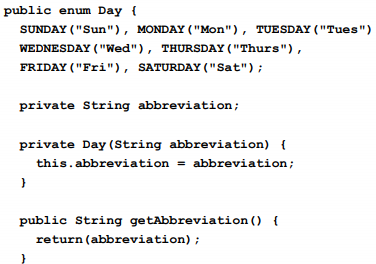
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | cl. | Pac. | subc | All |
| Pri | + |  |  |  |
| - | + | + |  |  |
| Prot. | + | + | + |  |
| Pub. | + | + | + | + |

**Final** -> it cnt be changed after init.

fin Cl -> cant ext. , fin f()->can’t override **Synchoronized**:mult. Th, could belocked **Volatile**: don’t cache, every th. Will get it from memory, will see update. Don’t use for race conditions. Doesnt solve it.**Transient**: It has meaning now, after transfer over network, it is meaningless.

**Native**: C/C++ imp. **@Override**-> cannot narrow visibl. Par.

Cl Par{ pub a()}. cl Ch. ex Par(@o Pr. a()}Cannot go public to private with @over. **Enums** enum E => cl Eext java.langEnu{} Enum doesn’t extend anything. Can have private/public members. **Constructors are private.**



When day is referred, **all constr**. will be called. Day d1 = Day.MONDAY, d1.get() .If your cl imp IA1,IA2, and those have same meth. You must override it. If cl ext s1, imp IA1, and those have same meth., super wins (w.o. override) Void d(Sh s)-> do is called **polymorphic. Generics** List<Emp.> workers=..;

Wor.add(new Emp()). Type check at comp. Time. After comp. No generic is present. Map<K,V> Map<Str, Emp>

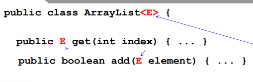
**Use Objects, not primitives!**

pub <T> T lastEl(List<T> ..), <T> says T is parameter, not the class name. Ex:

Integer[] nu={1,2,3}, lastEl(nu) OK

lastEl(1,2,3) OK. int[]num={1,2,3} NO.

Generics accepts object, not primitive. Use autoboxing.



AL<Person> bl = new AL<>; Inside <> can be empty. Compiler will add Per. İnside <>. You can narrow it by <? Extends Number>. Once generics are compiled, nothing left from generics in .class file. pub <T extends Number> T lastEl(List<T> ..), T is a Number or subclass.

Void f(SomeClass<?> sc){}, f() is not generic. It is accepting SomeCl. But Somecl is generic! Can limit by SomeCl<? Extends Number>

Pub cl Arrls<T extends Number>{

Pub T get(int ind){}}, in the Class, we define T ext Numb. We cant say T ext Num. İn get function. **Printf**

Syso(“Name %s”,name), %s is placeholder, not str,int etc. %f floating, %d wholenumb, %t time. String.format -> static. Printf(%8s”)

Printf(“%.2f”) -> 3.14. “\_\_\_\_\_\_Hi”

**Varargs** type... vars Any # of args will be accepted and put in vars[].

Only legal for last arg. Printf(Str format, Object... args). Ex:

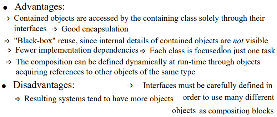
Pub stat. Void print(Obj... ent){...}

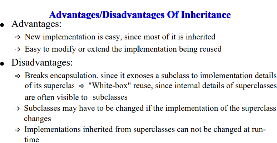
Print(1,2,3) ok. Integer[]={1,2,3} OK

İnt[] ={1,2,3} NO. It accepts object, not primitve. Ex2: Pub stat void ..(int... nums){}. Print(1,2,3)ok. İnt[]={1,2,3} OK. Integer[] ={1,2,3} NO. It accepts primitive array, not object array. **String Builder** str immutable, for loop içinde str’ye eklemek O(n\*\*2), strbuild’e eklemek o(n). **OO PRINCIPLES**

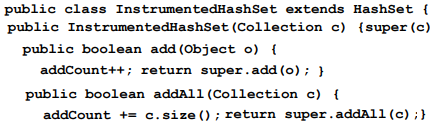
**1)Min. Acces. Of Class & members**

**2) Favor Composition over Inherit.**

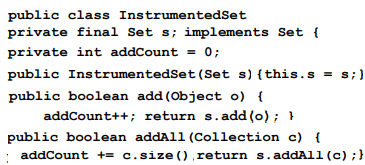




Ex of inheritance, when we add 3 element, count=6. Bc, addAll calls super AddAll(). Prob. Super.addAll calls add in forloop. Since InstHash. Extends that class, there is more special type of that add() method.

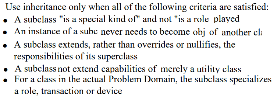


If InstSet implements Set interface, we need to write many functions but no side effects. Also, the InstSet can now have many different arguments in the constructor, bc. arg is Set and others impl. Set as well.

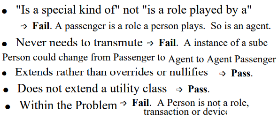




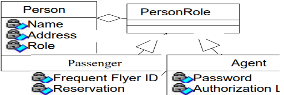
**Coad’s Rule**

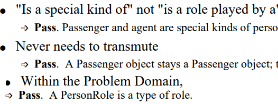


P <- Agent,Passenger <- AgentPass.

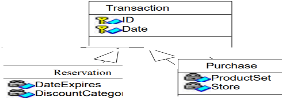


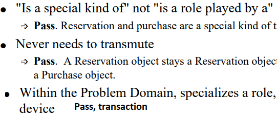
Use composition, Person has name, addr, pass, agent. This way OK.





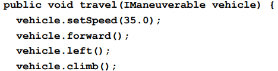
Ex:





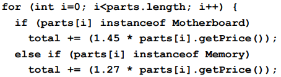
**3)Program To an Int. Not and Imple.**

2 class aynı interface i impl. edince, ikinsinde de aynı metodların olacağını biliyoruz. Aynı view’a sahip olacaklar. Methodlarda arguman olarak Class yerine Inter. alırsak, o inter.’i imp eden her class o met. sahip. Daha genel olur programımız.



**4) OCP,Open for ext. Closed for Mod.**

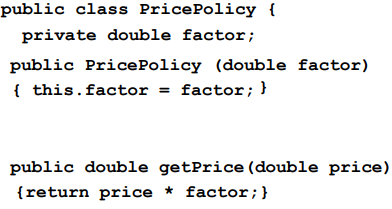
Not OCP code:

****

Also, not OCP code:

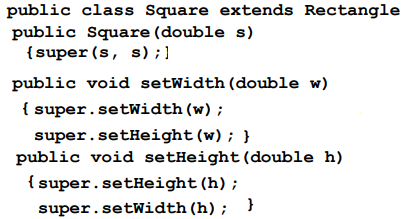
****

Instead, add pricepolicy. When the getPrice is asked ret pp.getPrice()

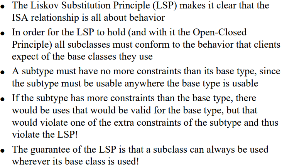


**Single Choice Principle:** If our system must support set of alter., write a config file, all altern. should stay there. 5)Liskov Substitution Principle:

Functions that use base class must use subclasses without knowing it.



Problems: programmer assumed that changing width not cause change in height. Also, setW,setH functions violates inheritance. They are overriding the existing funct. Not extending! So Square is not a Rect.



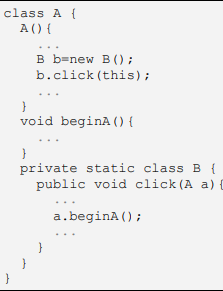
**PACKAGES**

x.y.A-> x can be anywhere in ClassPath. The first x.y.A will be taken. If you say no package name, cur. Directory. İf you say x.y.\*, the directory under x.y will not be taken (x.y.z.\* will not be imported).

İmport stat. Math.abs; Abs(x); -> w.o. Math prefix. Kendi başına duruyor, kimin methodu belli değil.

**INNER CLASSES**

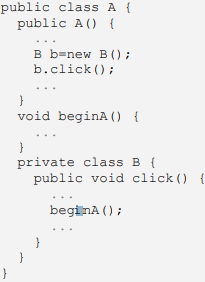
**Nested Class**



Normalde static clas init edilmez. Ama burada A’yı modify edebilir. Inıt şart. It will execute A through reference(this). Priv. Olduğu için dışarıdan erişilmez. Public olsa:

A.B b = new A.B(); gene de b.click()’in içine A referansı göndermek gerekir.

**Member Class**

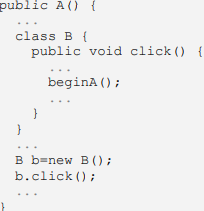


This olmadan beginA’ya access yapabilir. Each instance B will belong to outer class. Outer class can access member clas’ members/instances. Reverse is also true.

Member classes can't contain any static members, can't have the same name as any containing class or package. If the member class B public, then can be accessed from outside as:

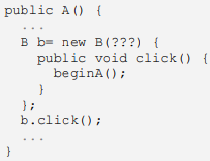
A a = new A(); A.B b = a.new B();

b. diyip A’nın metoduna erişemezsin. b.beginA() diyemezsin, b’de o yok. Ama b.click() diyebilirsin, A’ya erişebilirsin. **Local Class**

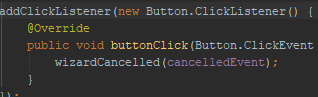


Local classes have the same rest. as member cl: local cl. can't contain any static members, can't have same name as any containing class or package. For all 3 class, if you compile the code, you’ll have A$B.class. B inside A. Sadece bu scope’ta geçerli. Ex: constr. dışına çıkınca B yok artık.

**Anonymous Class**



?? are params of the constructor. Overriding click method of Base class B or implementing click meth of Inter B. No name = no protection required. When compiled A$1.class



**Exceptions**: Throwable class. Error or exception. Cannot do anything to error. You must write try catch for exceptions. No need to write try catch for only RunTimeException.

Error->Out of mem., disk crash

If you throw and Exception, advertise it ...f() throws ClassNotFoundExc.

System checks try catch block from top to bottom. Write topmost specific, bottom Exception. **Finally**: it will be executed no matter what, even in return(not in sys.exit()). For housekeeping(closing file/socket). Writing own Exception: cl A ext RunTimeExc.{..f(){throw new A()}}

If assertion is not true, AssertionError is thrown. Assert(age>0): age2, can have the value in age2 for diagnose.

**Reflection CLASS**

\*Class c3 = obj.getClass( );

\*Class c=Class.forName( "j.u.Date");

Methods in Class:

\*String getName( ), \*boolean isInterface( ),\*boolean isArray(), \*Class getSuperclass( ),\*Class[] getInterfaces, \*Class[] getClasses( ) ->inner classes,\*Object newInstance(), \*static Class forName( String name ); \*Constructor[] getConstructors( );

\*Method[] getDeclaredMethods( ) -> public and private methods in the current class or interface

\*Method[] getMethods( )->public methods current and super class/interface. Ex:

c1 = Class.forName( "Hero" );

Class clArray[] = c1.getInterfaces( );

Interface: java.lang.Comparable

Constructor[]cA = c1.getConstructors( ); Constructor: Hero x2

Method[]mA=c1.getDeclaredMethods( ); Method: compareTo, getName, setName, restoreStrength(it is priv.)

Class p[] = mA[0].getParameterTypes( ); -> java.lang.Object. Hangi

parametreleri alacağını söylüyor.

Field f = aHero.getClass( ).getDeclaredField( “strength” ); Burada strength field’i public. Get/set yapabilir. F.get(Hero) -> 12, strengthini döndü. F.set(Hero, new Integer(10)). Eğer static bir field olsa Hero yerine null gelecekti. Ex:  
f.set(null, new Integer(10)). Obje olarak gönderiyoruz. New int[]değil.

\*Method getMethod( String methodName, Class[] paramTypes ); Returns a Method object that reflects the method identified by name and parameter types in the current class and all superclasses. Method must be public.

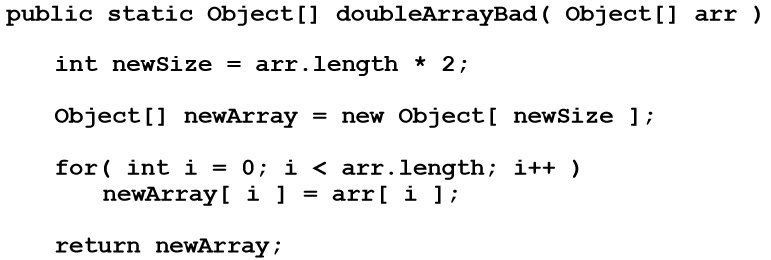
\*Method getDeclaredMethod( String methodName, Class[] paramTypes ); Returns a Method object that reflects the method identified by name and parameter types in the current class. Method may be private. **ARRAY**

All methods are static.

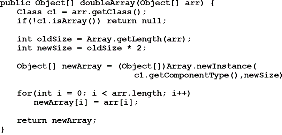
\*public int getLength( Object arr ); \*public Object newInstance( Class elements, int length ); \*public Object get( Object arr, int index )\*public void set( Object arr, int index, Object val ); \*public int getInt( Object arr, int index ); \*public void setInt( Object arr, int index, int val ). Creating Arr:

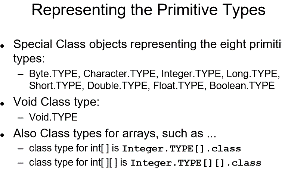
Class c1 = Class.forName( "Canine" ); Canine kennel = (Canine[]) Array.newInstance( c1, 10 );

Ex: Double Arr.

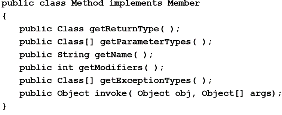


Won’t work. Because it accepts object array and copies objects to the new array. It doesn’t preserve types.

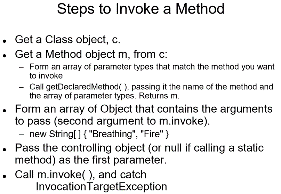


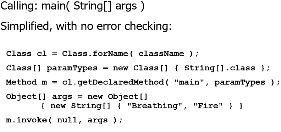


**METHOD**

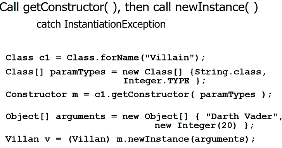


Obj[] args -> Integer.TYPE in an array





**Invoking Constructor**



\*Rectangle rect; \*Class rectDef;

\*Class[] intArgsClass = new Class[] {int.class, int.class};

\*Integer height = new Integer(12);

\*Integer width = new Integer(34);

\*Object[] intArgs = new Object[] {height, width};

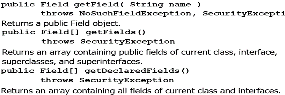
\*Constructor intArgsConst;

\*RectDef = Class.forName("..Rect");

\*intArgsConst = rectDef.getConstructor(intArgsClass);

\*rect = (Rectangle) intArgsConst.newInstance(intArgs);

**Field**



**KORAY YURTSEVEN,**

**2099547**



Hero h1 = new Hero( "Superman", "Clark Kent" );

Syso( "h1 = " + h1.toString( ) );

try {

Class cl = Hero.class;

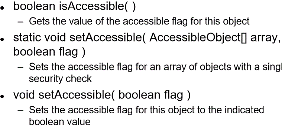
Field f = cl.getDeclaredField( "secretIdentity" ); // Secret identity is private, must set it accessible

f.setAccessible( true );

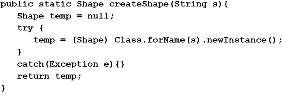
Syso ( "Secret identity = " + f.get(h1));

f.set( h1, new String( "Lois Lane") );

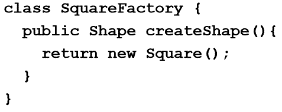
Syso( "New identity = " + f.get( h1 ) );



**Factory Methods** Bad:

****

Instead, create a Factory object for all types of instances, 1 time.



Shape square=squareFactory. createShape();. You can create Factory objects with reflection too.

**MULTITHREAD PROGRAMMING**

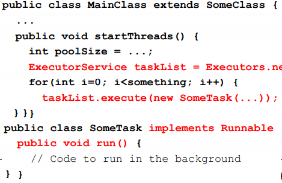
ExecutorService taskList = Executors.newFixedThreadPool(poolSize);

taskList.execute(someRunnable);

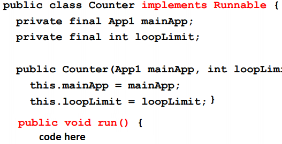
We can use NewCachedThreadPool(), ScheduledTP(size),SignleThExecutor(), SingleThScheduledExecutor()

To execute a class, it must implement Runnable

**Approach 1: Seperate Class**



Example SomeTask class:



taskList.execute(new Counter(this, 6));

taskList.execute(new Counter(this, 5));

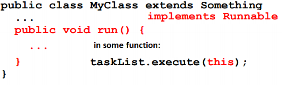
Adv/Disadv:

+ Can change independently

+ Can pass args to SomeTask.

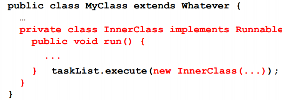
- Race cond. – Hard do access main app(use this reference).

**Approach2:Main App Imp. Runnable**

****

Can run methods in this class, since it is in this class. Cant customize(no constructor, only this. Aynı tL.exe(this) yazarak çalıştırıyoruz. Run metod sadece bu classta, bu class’a özel. Race conditions.

**Approach3: Inner Cl. Imp. Runnable**

****

Or you can use this too:

****

Can pass args. Easy to acces main app. Methods in inner class can acces any public/private methods in outerclass. –Race cond. –Run method tied to this app.

**Approach4: Lambda**

taskList.execute(()-> doSomeTask(..));