Java Essentials

Mastek

MASTEK INSTITUTE
FOR LEARNING & DEVELOPMENT

Evolution of Java

Java

- was created in 1991
- by James Gosling et al. of Sun Microsystems.
- Initially called Oak, in honor of the tree outside Gosling's window, its name was changed to Java because there was already a language called Oak.

Mastek

What is Java?

A high-level programming language

A development Environment

An Application Environment

A deployment Environment

Mastek

Java: Programming Language

As a programming language, Java can create all kinds of applications that you could create using any conventional programming language.

Mastek

Java: Development Environment

As a development environment, Java technology provides you with a large suite of tools:

- A compiler (javac)
- An interpreter (java)
- A documentation generator (javadoc)
- A class file packaging tool and so on..

Mastek

Java: An Application Environment

Java technology applications are typically general-purpose programs that run on any machine where the Java runtime environment (JRE) is installed.

Mastek

Java: Runtime Environment

There are two main deployment environments:

- The JRE supplied by the Java 2 Software Development Kit (SDK) contains the complete set of class files for all the Java technology packages, which includes basic language classes, GUI component classes, and so on.
- The other main deployment environment is on your web browser. Most commercial browsers supply a Java technology interpreter and runtime environment.

Mastek

Java Open or Closed?

Java is not quite an open language but not quite a proprietary one either

- compiler, virtual machines (VM), class packages, and other components - are free.

Java Community Process

- Leads the development of new standards for the language

Mastek

Editions of Java

Java Standard Edition (Java SE)

- A core set of components to create simple applications and support the advance applications

Java Enterprise Edition (Java EE)

- It provides a wide array of tools for building middleware software such as for database access applications, online storefronts, and other services.

Java Micro Edition (Java ME)

- J2ME replaces the Java 1.1 based systems
- The developer will choose from different configurations to suit the capacity of a given system.
- Specifically for the mobile applications developed in Java

Mastek

Java Development Kit

Java Runtime Environment (JRE)

- Java Virtual Machine (JVM)
- Java API: basic language + standard class library

Byte code compiler

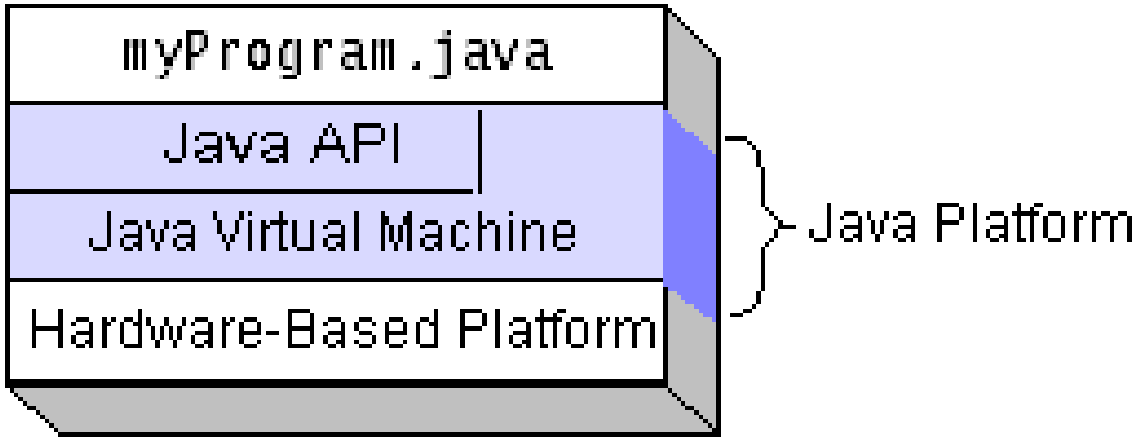
Other compilers, tools and utilities: debuggers, applet viewer, javadoc, RMI compiler, etc.

Mastek

Java platform components

The Java Virtual Machine (Java VM)

The Java Application Programming Interface (Java API)



Mastek

First Java Program

Introduction to Editor

Creating first Java program

Compiling the Java Program

Executing the Java Program

Mastek

Java Features

The Java Virtual Machine

Code Security

Garbage Collection

Mastek

The Java Virtual Machine

Java Virtual Machine (JVM)

- an imaginary machine that is implemented by emulating software on a real machine
- provides the hardware platform specifications to which you compile all Java technology code

Bytecode

- a special machine language that can be understood by the Java Virtual Machine (JVM)
- independent of any particular computer hardware, so any computer with a Java interpreter can execute the compiled Java program, no matter what type of computer the program was compiled on

Mastek

Code Security

Code Security is attained by Java through the implementation of Java Runtime Environment

JRE

- runs code compiled for a JVM and performs class loading (through the class loader), code verification (through the bytecode verifier) and finally code execution

Mastek

Code Security

Class Loader

- responsible for loading all classes needed for the Java program
- adds security by separating the namespaces for the classes of the local file system from those that are imported from network sources
- After loading all the classes, the memory layout of the executable is then determined. This adds protection against unauthorized access to restricted areas of the code since the memory layout is determined during runtime

Mastek

Code Security

Bytecode verifier

- tests the format of the code fragments and checks the code
- fragments for illegal code that can violate access rights to objects

Mastek

Garbage Collection

Garbage collection thread

- responsible for freeing any memory that can be freed. This happens automatically during the lifetime of the Java program.
- programmer is freed from the burden of having to deallocate that memory themselves

Mastek

Data Types and Variables

Mastek™

Primitive and Reference Types

Each data value has a type

The type must be declared for all variables & constants

The compiler needs this information

- to allocate memory for the variable or constant
- to verify that the variable/constant is being used correctly

Mastek

Primitive Types

Also known as "simple types"

- int, byte, short, and long for integer values
- float and double for real/fractional values
- char for letters, digits, symbols, punctuation
- boolean for true and false

Literals:

- values of one of these types
- 'A', -7, 3.5, true

Mastek

Fields vs. Local Variables

Fields are declared outside all constructors and methods.

Local variables are declared inside a constructor or a method.

Fields are usually grouped together, either at the top or at the bottom of the class.

The scope of a field is the whole class.

Mastek

Reference Types

Needed to represent windows, buttons, inventory, students, etc.

- objects more complicated than can be represented with simple types

Create a class and you have created a new type whose name is the name of the class

Types created from classes are called reference types

Mastek

Default Value: null

Default value for reference types

```
Screen theScreen = null;    //or
Screen theScreen;
```

- value used if none is specified in declaration
- indicates variable does not yet refer to a value of that type
- can later be assigned values created with new

Mastek

Wrapper Classes

Primitive Types	Wrapper Classes
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

Mastek

Operators in Java

Mastek™

Numeric Expressions

Primitive Expression

- sequence of one or more operands (values)
- combined with zero or more operators
- result is a value
- the type of the value produced is the type of the expression (int, double, etc.)

Example

2 + x * (3 + y)

Mastek

Operators

Operator Type	Operators
Arithmetic	+, -, *, /, %
Unary	++, --
Assignment	+=, -=, *=, /=, %=
Relational	<, >, <=, >=, ==, !=
Logical	&&, ||, !
Conditional	<condition>?<true>:false

Mastek

Control Structures

Mastek™

Selection statements

if-else

```
if ( <condition> ){
        <statements >
} else if(<condition>){
        <other statements
>
} else{
        <other statements
>
}
```

switch

```
switch  (expression)
{
case value1: <statement>
        break;
case value2: <statement>
        break;
default: <statement>
}
```

Mastek

Iteration statements

while

```
while ( condition ){
statement1;
statement2;
...
statementN;
}
```

do-while

```
do {
statement1;
statement2;
...
statementN;
} while ( condition );
```

for

```
for ( initialization;
condition; increment
)
{
statement1;
statement2;
...
statementN;
}
```

Mastek

Jump statements

break

continue

return

Mastek

Arrays

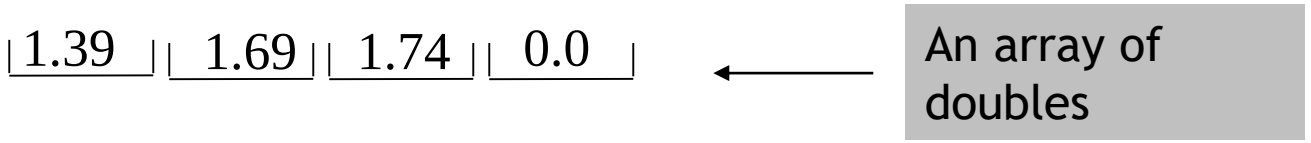
Mastek™

What is an Array

An array is a block of consecutive memory locations of the same data type.

Individual locations are called array's elements.

Sometimes when we say "array's element" we mean the value stored in that element.

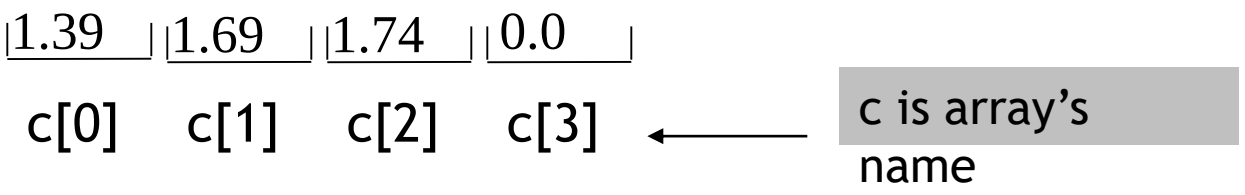


Mastek

What is an Array (cont'd)

Rather than treating each element as a separate named variable, the whole array gets one name.

Specific array elements are referred to by using array's name and the element's number, called index or subscript.



Mastek

Indices (Subscripts)

In Java, an index is written within square brackets following array's name (e.g., a[k]).

Indices start from 0; the first element of an array a is referred to as a[0] and the n-th element as a[n-1].

An index can have any int value from 0 to array's length - 1.

Mastek

Arrays as Objects

In Java, an array is an object. If the type of its elements is anyType, the type of the array object is anyType[ ].

There are two ways to declare an array:

```
anyType [ ]  arrName;

    or

anyType  arrName [ ];
```

The difference becomes significant only when several variables are declared in one statement:

```
int [ ] a, b;    // both a, b are arrays
int a [ ], b;    // a is an array, b is not
```

Mastek

Declaration and Initialization

When an array is created, space is allocated to hold its elements. If a list of values is not given, the elements get the default values.

```
scores = new int [10] ;
                    // length 10, all values set to 0

words = new String [10000];
                   // length 10000, all values set to null
```

Mastek

Object Oriented Programming

Mastek™

Object Oriented Programming

A system that is made up of units that holds together data and the functionality associated with data, is said to be object oriented.

Object Oriented Computing is a different way of looking at the world

“Anything in this world can be automated”

Mastek

OOP Concepts

Abstraction

Encapsulation

Polymorphism

Inheritance

Mastek

Abstraction

Abstraction is a purposeful suppression, or hiding, of some details of a process or an artifact, in order to bring out more clearly other aspects, details, or structure

Modeling of real world objects makes it easier to describe and communicate behavior

- Level of abstraction depends on the requirements

Mastek

Encapsulation

Encapsulation is a method which is used to hide the working details of an entity

Encapsulation is meant to prevent direct access to classified data

Security of data members and behavior plays important role in any system

Mastek

Messaging

Messaging is the method of communication between the units that comprise an object oriented system

A message (method call) has four parts

- Identity of the recipient object
- Code to be executed by the recipient
- Arguments for the code
- Return value

Mastek

Inheritance

Inheritance is a method that captures the commonality

This is a relationship between units where one unit inherits all or a part of description of other more general unit

It elaborates “is a kind of” relationship

Mastek

Polymorphism

Polymorphism is the ability to appear in different scenarios

It refers to the ability to process objects differently depending on their data or type

Different types of polymorphism

- Compile-time polymorphism (Overloading)
- Runtime polymorphism (Overriding)

Mastek

Classes and Objects

Class represents an abstraction of real world. Class is the blue print of a set of objects.

Related data and procedures (methods) are grouped together in a class

Portion of a class is private and is hidden from client users of the class (encapsulation).

Rest can be accessed outside the class.

Mastek

Classes and Objects

A class is a template that describes the data structure and methods that operate on that data

```
public class Car {
        Color car_color;
  void run ( ) {  }
  }
```

An object is an instance of a class

```
Car c =  new Car ( );
```

All the objects are created dynamically

Mastek

Accessing data and methods

Members of an object are accessed by

object_name.member_name

Accessing object's data

```
Car c = new Car ( );
c.car_color = Color.red;
```

Using an object's method

```
c.run ( );
```

Mastek

Objects

Elements

- private data field
- public field accessor/getter method
- public field mutator/setter method

Benefits

- easy change in internal representation
- mutator can provide error checking

Mastek

Objects

Accessor/Getter Method

- getting the object's state

Mutator/Setter Method

- changing the object's state

Constructors

- has the same name as the class
- may have parameters
- called with new keyword

Mastek

Constructor

Constructors do not return anything, not even void

```
class Complex {
 double real, imag;
 public Complex (double r, double i)
    { real = r; imag = i; }
}
Complex c = new Complex(1.0, 2.3);
```

Mastek

The this reference

As in C++ the reserved word this represents the current object

Object's methods will have this as an implicit first parameter

In C++ this is a pointer, so we can use

this->car_color

In Java we use

this.car_color

Mastek

static variables

Java does not have global variables

Every variable must be written inside a class

The static variables can be used to share a variable class-wide

Such a variable is called a class variable

```
public  class Math {
    static double final PI = 3.141596;
}
```

Mastek

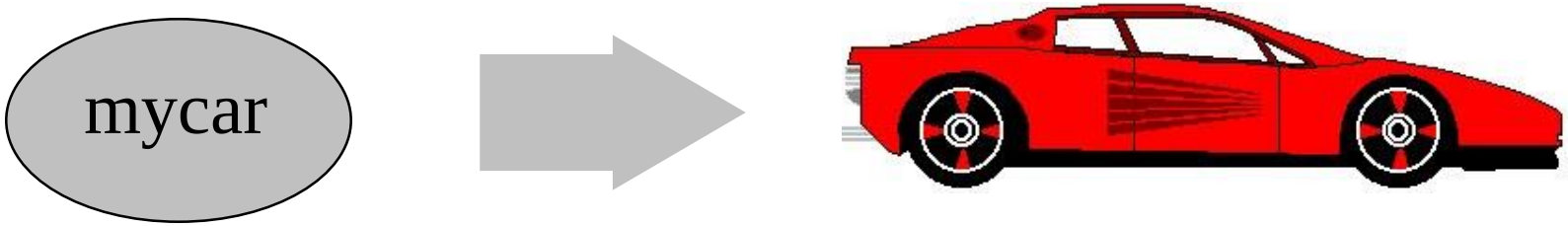
Object Creation

Car mycar

does not refer to any object

cannot use methods in Car

Car mycar = new Car( ) ;

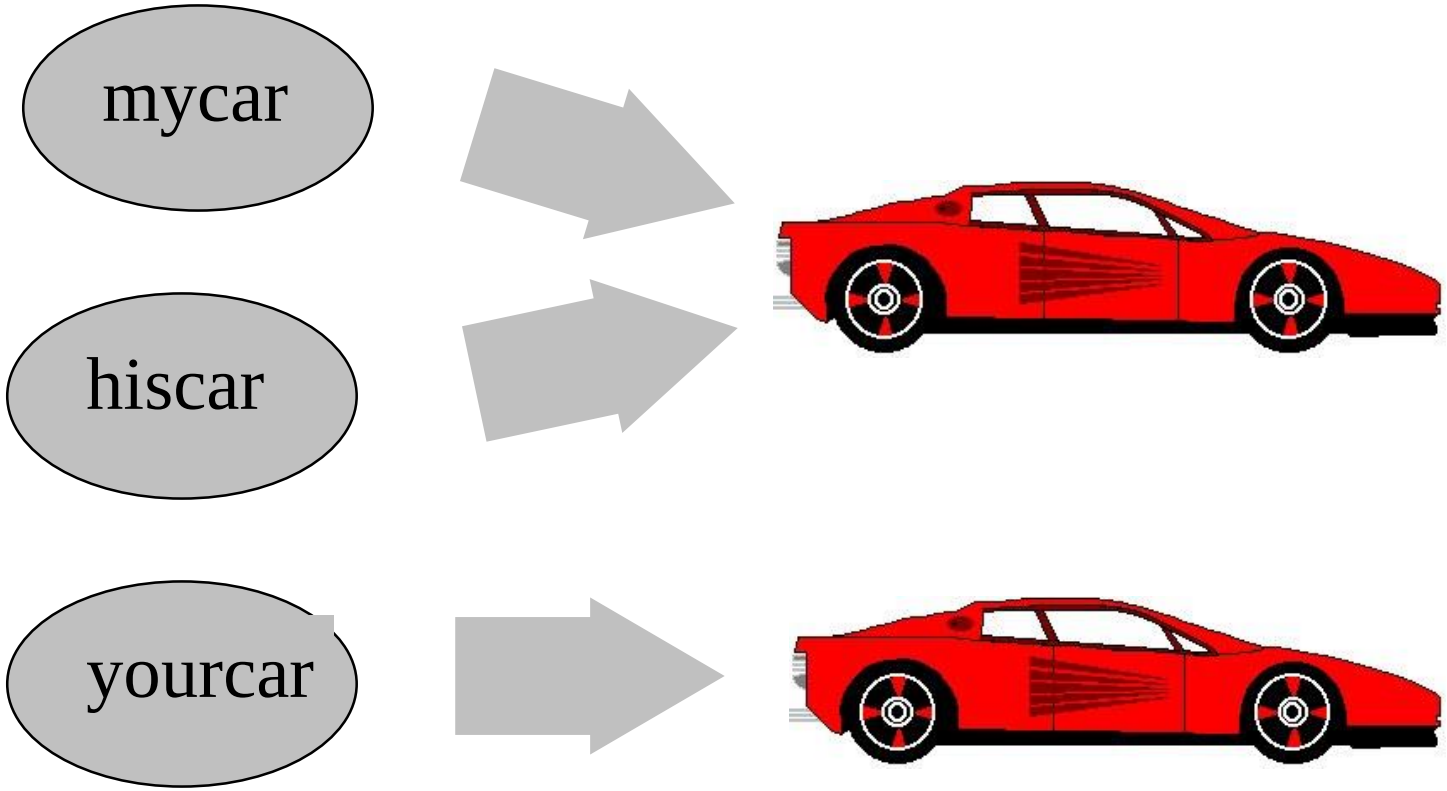


Mastek

Object Creation

Car yourcar = new Car( );

Car hiscar = mycar



Mastek

Overloading

A class can have multiple constructors, if the number and/or type of parameters vary

Constructor is said to be overloaded

Any method can be overloaded in Java

```
class complex { double real, imag;
public complex ( )
    { this( 0.0, 0.0); }
public complex( double r, double I)
    { real = r ; imag = I ; } }
```

Mastek

Blocks

Instance Block for object Creation

- { <any code>}

Static Block for Class Loading

- static{ <any code for static initialzer> }

finalize() method for object destruction

Mastek

Object Lifecycle

1. On Execution of the Application [for Each Class]
    a. static variables are initialized
    b. static block is executed
2. On Each Object creation using 'new'
    a. Object is created the heap memory
    b. instance variables are initialized
    c. instance block is executed
    d. The Constructor is Executed
    e. Object is ready to use
3. On Garbage Collection
    a. Identifies the unreferred Object [Garbage]
    b. Executes the finalize method for Each Object to be deleted
    c. Deletes the Object from Memory

Mastek

Inheritance

Mastek™

Inheritance

Inheritance is a method that captures the commonality

This is a relationship between units where one unit inherits all or a part of description of other more general unit

It elaborates “is a kind of” relationship

Mastek

Inheritance

Inheritance implements is-kind-of relationship between classes

(e.g. Car is a kind of vehicle)

Any class can extend any other class

Inheritance is a mechanism for code reuse

All the classes in Java are arranged in a single inheritance hierarchy

The root of all classes is called Object

Mastek

Inheritance

Polymorphism means creating a method in a derived class that has the same signature as the method in the base class

All methods are virtual as opposed to C++

Methods can be non-virtual by declaring them as final

A class can be made non-subclassable by declaring it as final

Mastek

Abstract Classes & Interfaces

Mastek™

Abstract Classes

Some methods can be declared as abstract

Derived class should override these methods

(pure virtual member functions of C++)

A class having one or more abstract methods is an abstract class.

Abstract classes cannot be instantiated

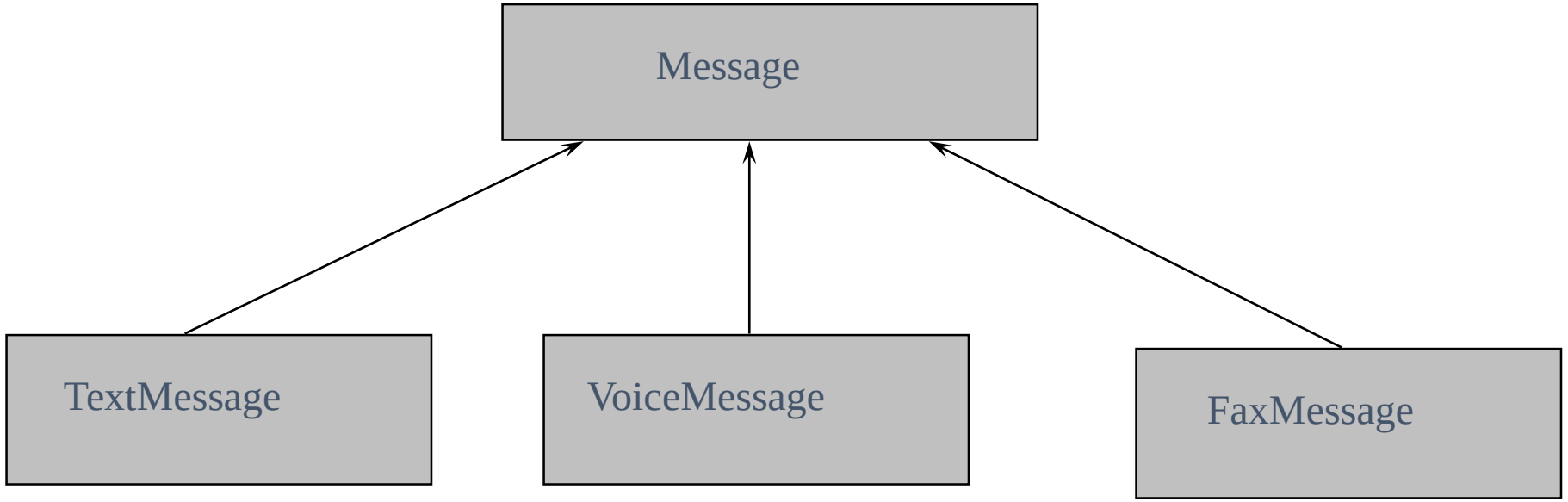
Mastek

Abstract Classes

Classes become more and more general

Method cannot be specified in a class

Abstract classes have at least one abstract method



Mastek

Interfaces

Interface is a group of methods just declared but not implemented.

All the methods of the interface are abstract.

Any class can implement any interface

An interface cannot be instantiated

Interface can have data members.

But all data members have to be static public final

Mastek

Interfaces

We can simulate multiple inheritance using interfaces.
Interface is a type.

Class is an implementation of one or more types

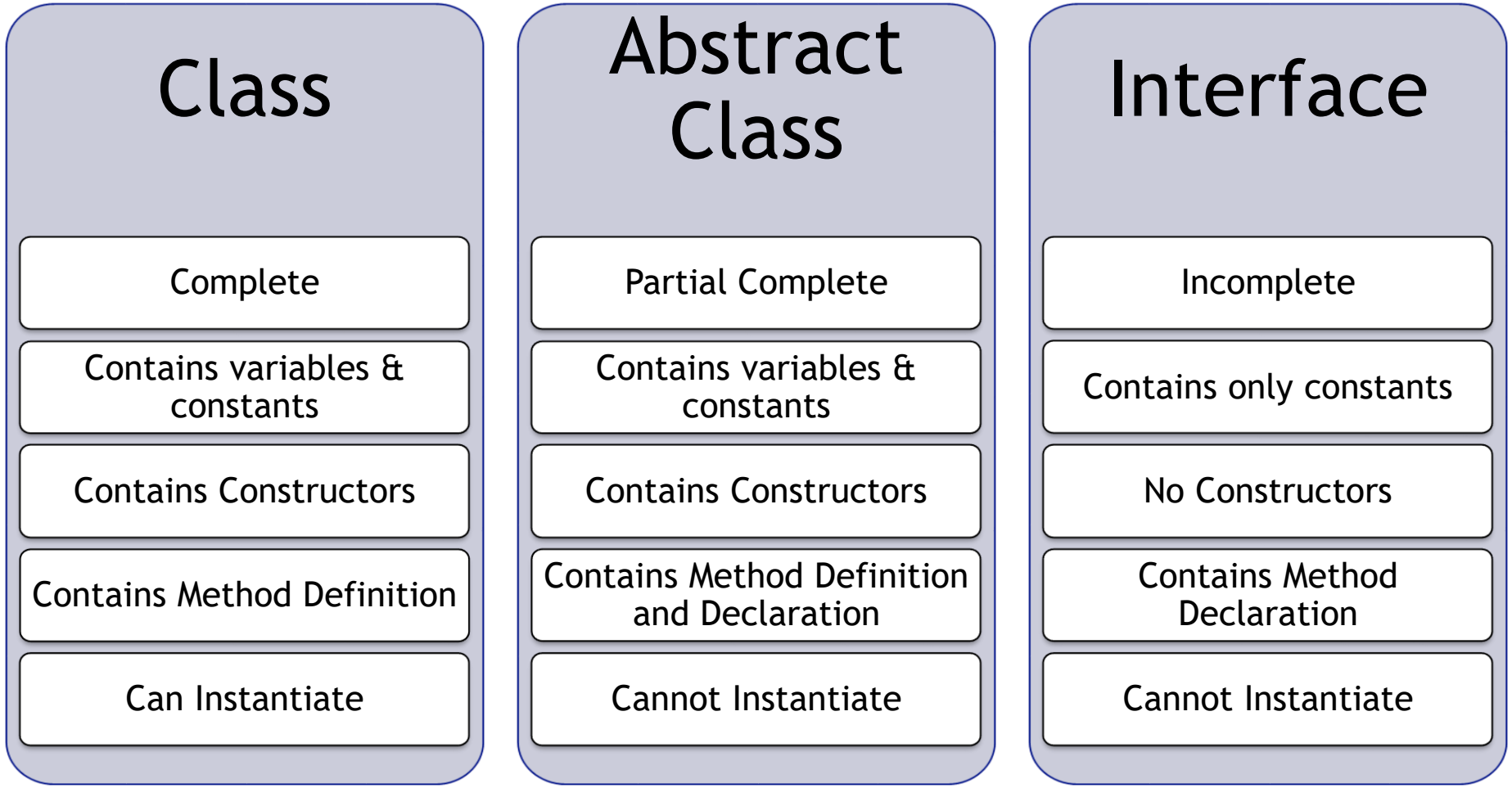
Interface can extend one or more interfaces

Interface can also be used as operand of instanceof operator

Implementation inheritance Vs interface inheritance

Mastek

Class, Abstract Class & Interfaces



Mastek

Packages

Mastek™

Package

There is package level information hiding.

Classes and interfaces can be public and available to other packages or may be internal to a package

To refer to a class or a interface of other package, we can use the syntax

package_name.class_name or

package_name.interface_name

This way we can avoid import declaration

Mastek

Package Syntax

A package is declared by

```
package package_name;
```

This should be the first statement in the file

There can be at most one public class or interface in a file

In a file we can import classes in other packages by

```
import java.io.*;
import java.awt.*;
```

Mastek

Access Control

public: can be used anywhere

private: can be used only by methods of the same class

protected: can be used by subclasses in other packages and by all the classes in the same package

<empty>(default): can be used only by classes of the same package

Mastek

Access Control Matrix

Scenario	Public	Protected	Default	Private
Same Class	☑	☑	☑	☑
Another Class Same Package	☑	☑	☑	☒
Derived Class Same Package	☑	☑	☑	☒
Derived Class Another Package	☑	☑	☒	☒
Another Class Another Package	☑	☒	☒	☒

Mastek

Exception Handling

Mastek™

Exceptions

Exceptions are unusual things that happen within your Java program that are different from the desired behavior of the program.

They could be fatal errors or could be in the event of exceptional circumstances.

Exception handling is the management of these exceptions or errors.

When an exception is encountered, Java will report this problem by throwing an exception.

Mastek

Exceptions

At this point, the system will halt normal operation and look for a solution to the problem.

It looks in your code, to see if there is anything that will catch the exception.

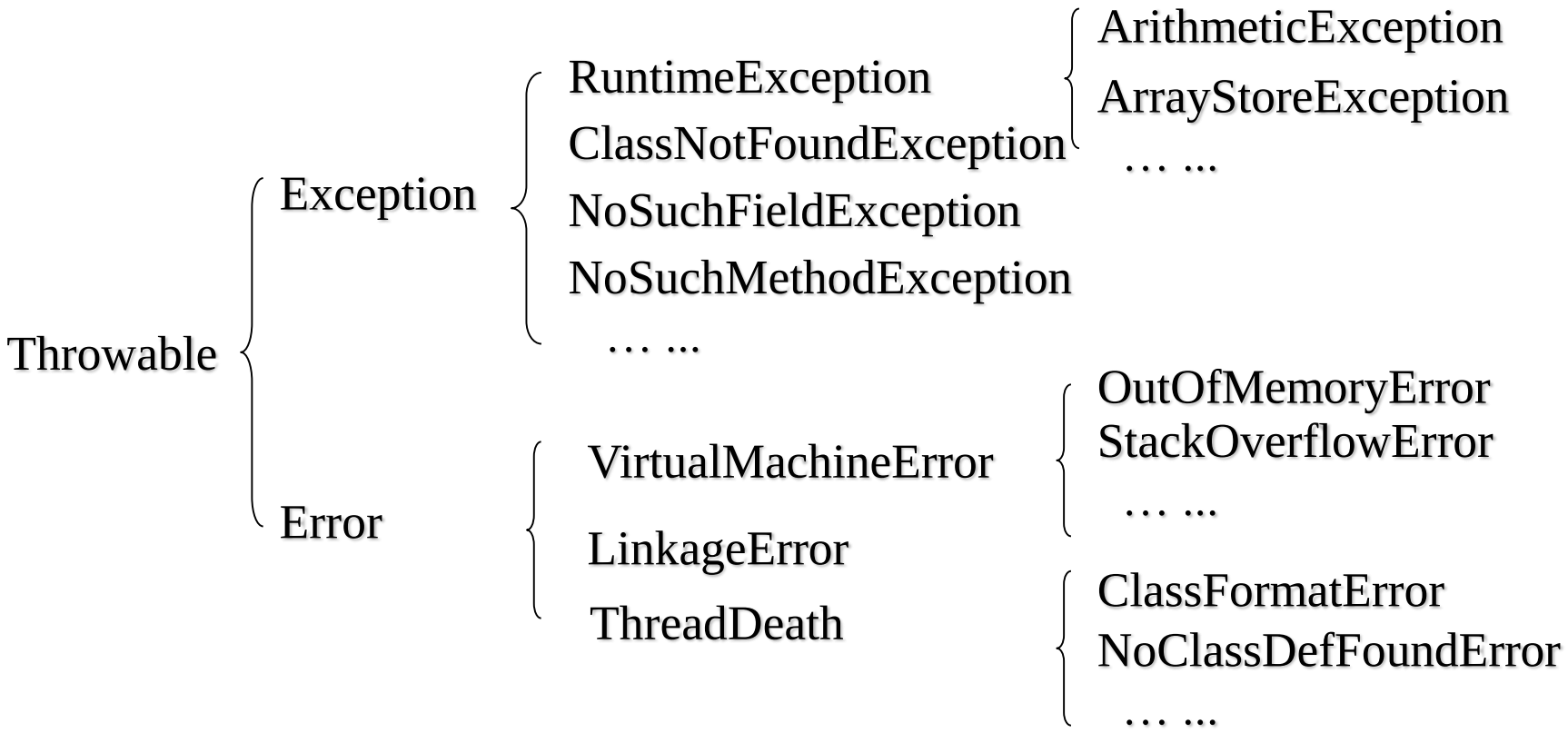
After the exception is caught, normal operation is resumed after the offending block of code.

In traditional error handling, some code is usually written to head off the error before it occurs.

Mastek

Exceptions

Exceptions in Java are actual objects of classes that inherit from the Throwable class.



Mastek

Throwable Class

The Throwable class has two subclasses: Error and Exception .

Instances of the Error class represent internal errors that are usually fatal. You can neither catch them nor throw them yourself.

The Exception subclass will contain most of the exception classes that are used, but there are other packages that define their own exception classes.

For example, the java.io package has its own exception class called IOException.

Mastek

Try and Catch

```
try {
        offset = x / n;
        // anything from here down will be ignored if n is 0.
}
catch (ArithmeticException e){
                offset = 10;
}
// Execution will continue from here after the exception is
handled
```

Mastek

Multiple Catch Clauses

catch clauses are examined from top to bottom, stopping at the first clause that has an argument that is compatible with the thrown exception and then skipping the remaining clauses.

```
try {
        …
}
catch (Exception e) { // Compatible with every exception
A       …
}
catch (ArithmeticException e) { // will never be called
                …
}
```

Mastek

finally Clause

Suppose that there is some code that must be executed whether an exception was thrown or not. This is done in what is called the finally clause.

```
SomeFileClass f = new SomeFileClass();
    if (f. open(“/ a/ file/ name/ path”)) {
            try {
            someReallyExceptionalMethod();
            } catch (IOException e) {
            // deal with them!!
            } finally {
                    f. close();
            }//finally
    }//if
```

Mastek

The throws Clause

To indicate that some code in the body of your method may throw an exception, use the throws keyword after the signature of the method.

```
public boolean myMethod( int x) throws AnException { … }
```

For multiple exception types, put them all in the throws clause separated by commas.

Mastek

The throw Clause

The simplest way is to create the instance and throw the exception in the same statement.

- throw new ServiceNotAvailableException();

Depending on the exception class that you are using, the exception's constructor may require some arguments. The most common argument is a string that describes the exception in more detail.

- throw new ServiceNotAvailableException(" Exception: service not available, database is offline.");

Mastek

Creating Exception

Although Java provides a wealth of exception classes, there may be times where we would like to create our own exceptions that are not covered by the predefined exception classes.

To create a exception, you must inherit from one of the other exception classes.

Try to find an exception that’s close to the one you are creating.

Mastek

Exceptions Pros and Cons

Pros:

- cleaner code: rather than return a boolean up the chain of calls to check for exceptional case, throw exception
- lets you use return value for something meaningful beyond error checking
- error handling not mixed in with normal code

Cons:

- throwing exceptions requires a lot of computation
- can become messy if not used sparingly
- be careful not to catch exceptions such as NullPointerExceptions which might be better left uncaught

Mastek

Thank you

Mastek™