Getting Started with Java

Michael P. Redlich

@mpredli

Who's Mike?

- InfoQ Java Queue News Editor
- Co-Director, Garden State Java User Group
- Leadership Council, Jakarta EE Ambassadors
- Committer, Jakarta NoSQL and Jakarta Data
- "Petrochemical Research Organization"

Objectives (1)

- What is Java??
- Evolution of Java
- Features of Java
- Review of Object-Oriented Programming (OOP)

Objectives (2)

- Getting Started with Java
  - introduction to the Java class mechanism
    - how to implement Java classes
- Live Demos (yea!)
- Java Resources

What is Java?

- *"Java is C++ without guns, knives, and clubs."*

James Gosling, "father" of Java, Sun Microsystems

- *"Java is simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high performance, multithreaded dynamic language."*

Sun Microsystems

Evolution of Java (1)

- Created by James Gosling (with Patrick Naughton)
  - Sun Microsystems
- 1991 - originally named "`Oak`"
  - consumer applications
  - architecture agnostic
  - object-oriented

Evolution of Java (2)

- 1994 - Project “*7” dissolved
  - but shortly thereafter…
- 1995 - Java introduced at Sun World ’95
  - HotJava browser

Features of Java

- Object-Oriented Programming (OOP) Language
- Documentation Generation
- ~~Applets and~~ Applications
- Comprehensive Exception Handling
- Java Database Connectivity (JDBC)
- Java Beans
- Records/Sealed Classes
- Enterprise Java ~~Beans~~
- No pointers!!

OOP Review (1)

- Programming Paradigm
- Four (4) Main Attributes
  - data encapsulation
  - data abstraction
  - inheritance
  - polymorphism

OOP Review (2)

- Abstract Data Type (ADT)
  - user-defined data type
  - use of objects through functions (methods) without knowing the internal representation

OOP Review (3)

- Interface
  - functions (methods) provided in the ADT that allow access to data
- Implementation
  - underlying data structure(s) and business logic within the ADT

OOP Review (4)

Class

- Defines a model
- Declares attributes
- Declares behavior
- Is an ADT

Object

- Is an instance of a class
- Has state
- Has behavior
- May have many unique objects of the same class

Advantages of OOP

- Interface can (and should) remain unchanged when improving implementation
- Encourages modularity in application development
  - Better maintainability of code
  - Code reuse
- Emphasis on what, not how

Some Java Keywords

- `class`
- `new`
- `private`, `protected`, `public`, `package`
- `try`, `throw`, `catch`, `finally`
- `final`
- `extends`
- `implements`
- `abstract`
- `interface`

Classes (1)

- A user-defined abstract data type
- Extension of C `structs`

- Contain:
  - constructor
  - data members and member functions (methods)

Classes (2)

- Dynamic object instantiation
- Multiple Constructors:
  - `Sports();`
  - `Sports(String,int,int);`

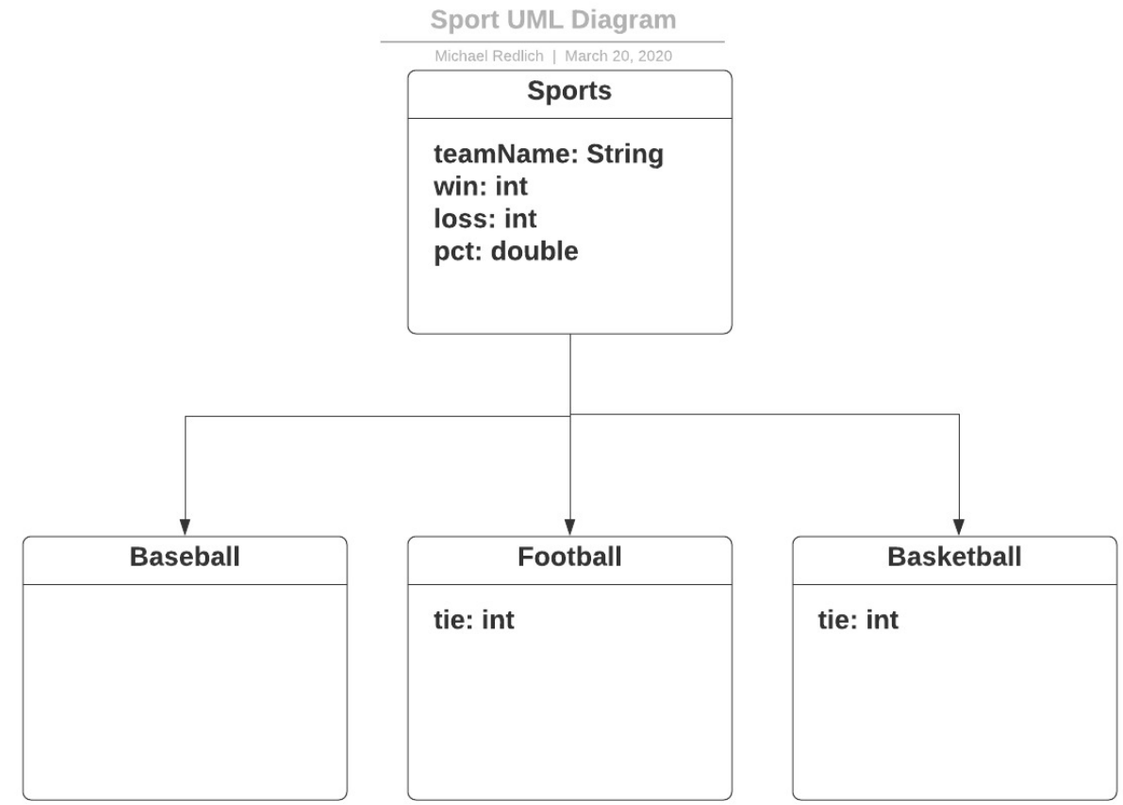
Classes (3)

- Abstract Classes
  - contain at least one pure virtual member function (C++)
  - contain at least one abstract method (Java)

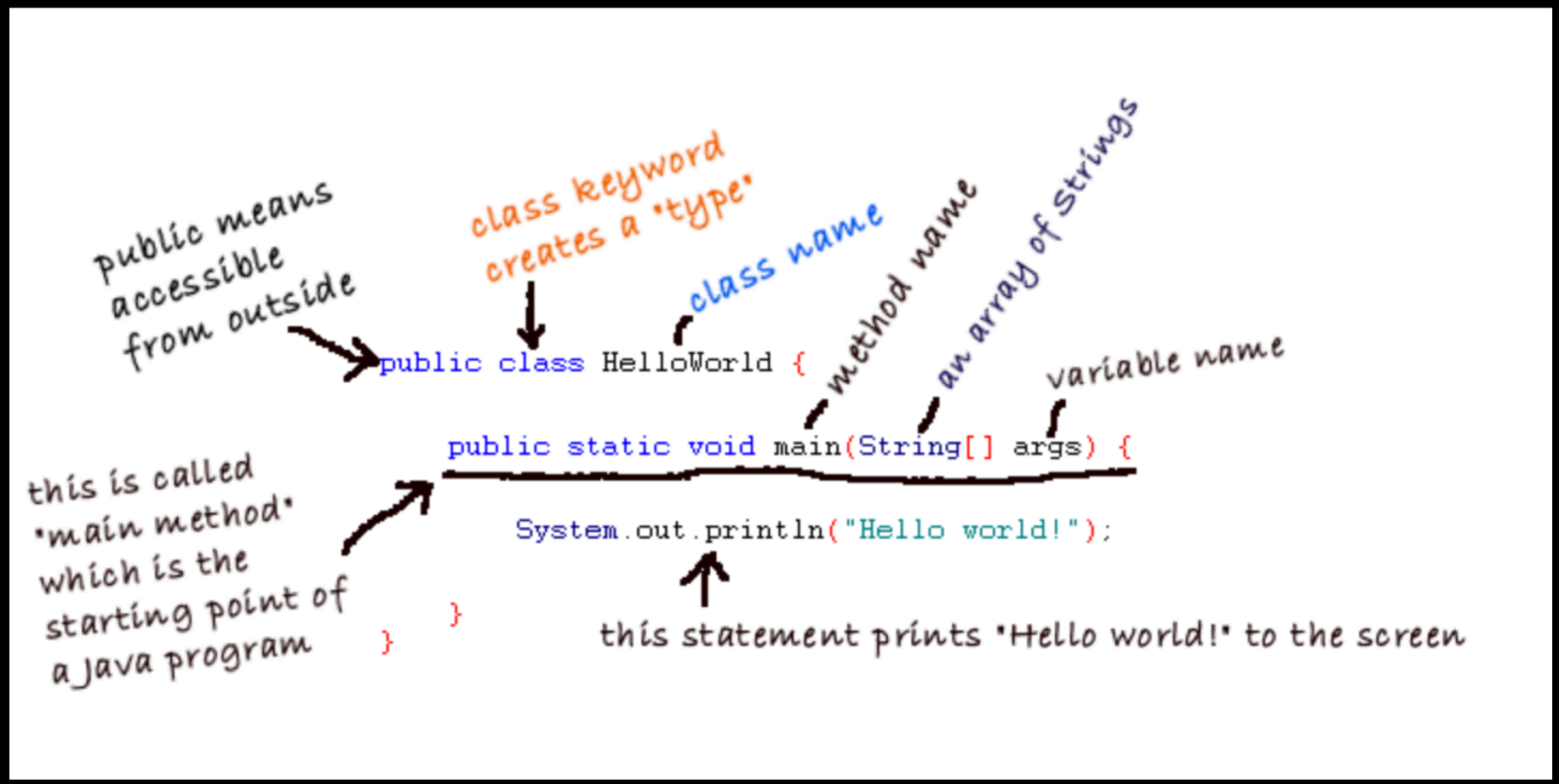
Abstract Classes

- Pure virtual member function (C++)
  - `virtual void draw() = 0;`
- Abstract method (Java)
  - `public abstract void draw();`

Class Inheritance



Hello, World!



source: http://www.codejava.net/java-core/how-to-write-compile-and-run-a-hello-world-java-program-for-beginners


```
// Sports class (partial listing)

public class Sports {
  private String team;
  private int win;

  public Sports() {
    // define default constructor here...
    }

  public Sports(String team,int win,int loss) {
    // define primary constructor here...
    }

  public int getWin() {
    return win;
    }
}
```


```
// Baseball class (partial listing)

class Baseball extends Sports {
  public Baseball() {
    // define default constructor here...
    }

  Baseball(String team,int win,int loss) {
    // define primary constructor here...
    }
}
```

Dynamic Instantiation

- Object creation:

```
Baseball mets = new
Baseball("Mets",97,65);
```

Deleting Objects


```
Baseball mets = new
Baseball("Mets",97,65);
// automatic garbage collection or:
System.gc(); // explicit call
```

Java Development Kit (JDK)

- Available from Oracle web site
  - `java.sun.com`
- Java SE (standard edition)
- Six-month release cycle since JDK 9
- latest version - JDK 20
  - released on March 21, 2023

Working with Java (1)

- Setup environment and path:
  - `set JAVA_HOME=path`
  - `set PATH=%PATH%;%JAVA_HOME%\bin`
  - `export JAVA_HOME=path`
  - `export PATH=$JAVA_HOME/bin`

Working with Java (2)

- Source code
  - `.java` extension
- Intermediate bytecode
  - `.class` extension generated after successful compilation
- Bytecode interprested by Java Virtual Machine (JVM)

Working with Java (3)

- Compile Java source code:
  - `javac -Xlint:all -d` *`path`* *`filename`*`.java`
- Run the application:
  - `java -classpath` *`path`* *`filename`*

Directories and Packages (2)

- Consistent directory structure
  - source code (`*.java`)
  - byte code (`*.class`)
- Map directories with package name
  - under the `src` folder
  - under `src/main/java` folder (Maven, Gradle)

Directories and Packages (2)


```
/usr/local/apps/java-apps
  ↳java-apps
    ↳tcf
      ↳hello
        ↳src
          ↳org
            ↳redlich
              ↳hello  ➜ package org.redlich.hello;
```


GARDEN STATE
GARDEN STATE
JAVA USER GROUP
NEW JERSEY
JAVA USER GROUP

Demo

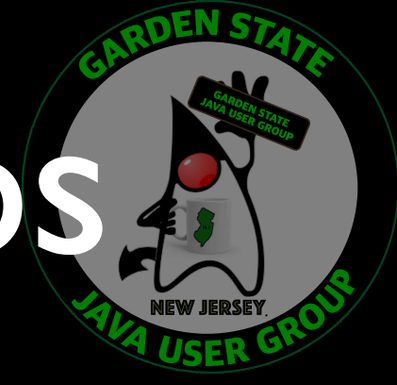
Java IDEs (1)

- IntelliJ IDEA 2022.3
  - `jetbrains.com/idea`
- Eclipse IDE 2023-03
  - `eclipse.org/eclipseide`

Java IDEs (2)

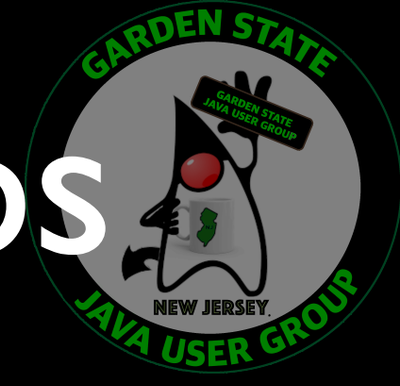
- Microsoft Visual Studio Code 1.76
  - `code.visualstudio.com`
- Apache NetBeans 17
  - `netbeans.apache.org`

Local Java User Groups (1)



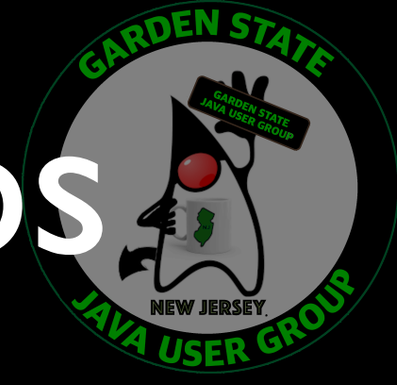
- Garden State Java Users Group (GSJUG)
  - facilitated by the GSJUG Leadership Team
  - `gsjug.org`
- NYJavaSIG
  - facilitated by Frank Greco, et.al
  - `javasig.com`

Local Java User Groups (2)



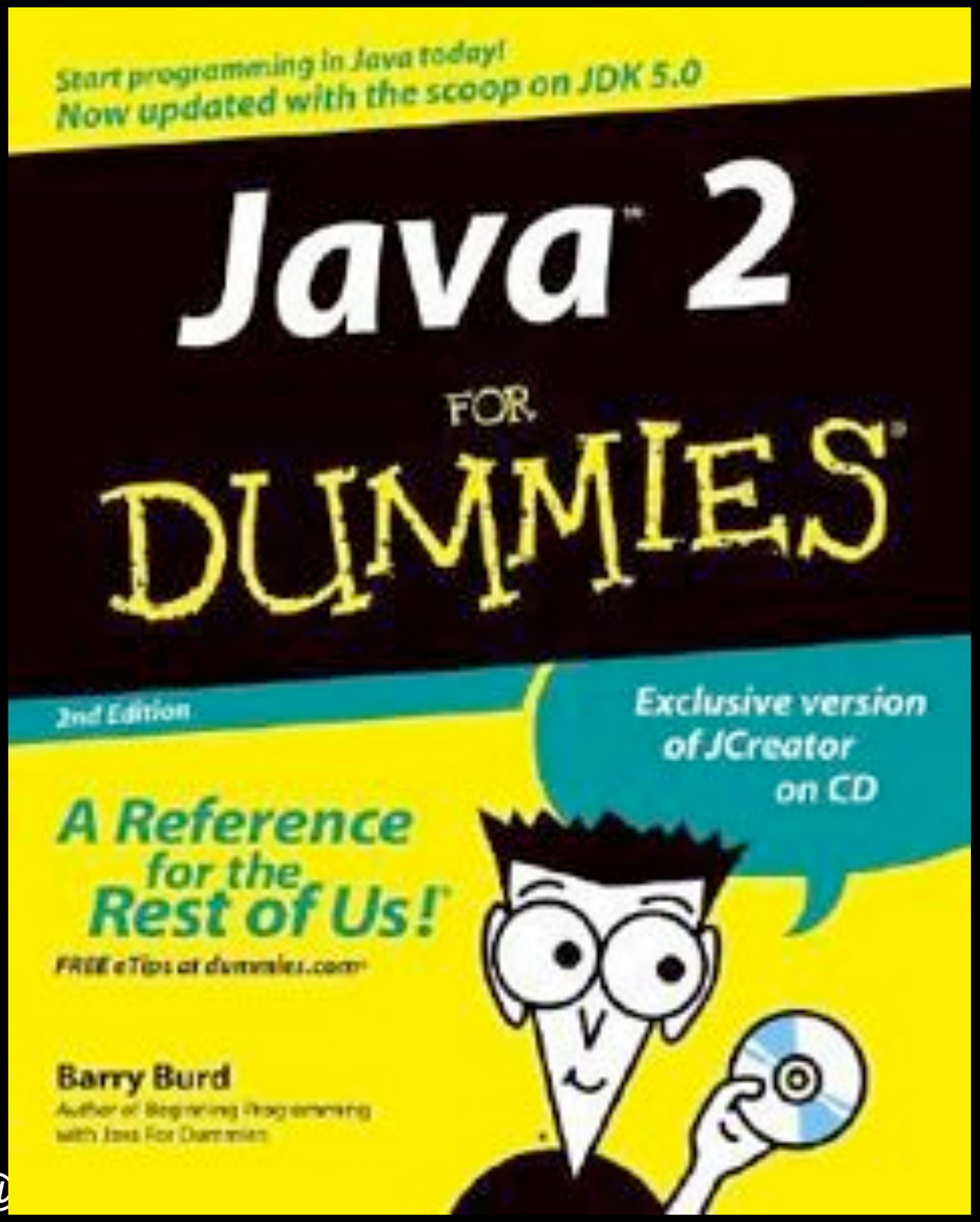
- PhillyJUG
  - facilitated by Paul Burton, et. al.
  - `meetup.com/PhillyJUG`
- Jersey City Java Users Group
  - facilitated by Amitai Schleier
  - `meetup.com/Jersey-City-Java-User-Group-JC-JUG/`

Local Java User Groups (3)



- Capital District Java Developers Network
- facilitated by Dan Patsey
- `cdjdn.com`
- currently restructuring

Further Reading



Contact Info

mike@redlich.net

@mpredli

redlich.net

redlich.net/portfolio

github.com/mpredli01

Thanks!

@mpredli