

2

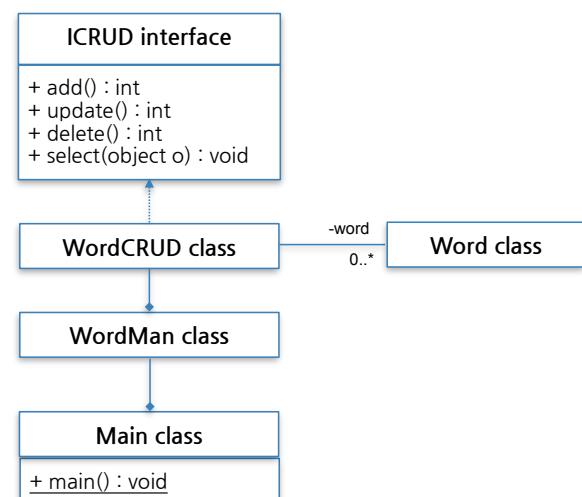
# Java CRUD 1

## Java CRUD - 프로젝트의 구성

- Word class 데이터클래스
- ICRUD interface CRUD를 위한 interface
- WordCRUD class ICRUD interface 구현체
- WordManager class WordCRUD를 사용한 데이터 처리
- Main class static main 함수, starter class

Dictionary.txt

```
3|superintendent|관리자, 감독관  
1|electric|전기의, 전기를 생산하는  
2|equipment|장비, 용품  
1|pole|기둥, 장대  
1|formerly|이전에, 예전에
```



# Java CRUD – Create

- WordManager class
  - selectMenu()
- WordCRUD class
  - Create : 새로운 Word 추가
  - Read : 등록된 모든 Word 조회
- Word class
  - `toString()` : overriding

\*\*\* 영단어 마스터 \*\*\*

\*\*\*\*\*

1. 모든 단어 보기
2. 수준별 단어 보기
3. 단어 검색
4. 단어 추가
5. 단어 수정
6. 단어 삭제
7. 파일 저장
0. 나가기

\*\*\*\*\*

=> 원하는 메뉴는?

Create

=> 원하는 메뉴는? 4

=> 난이도(1,2,3) & 새 단어 입력 : 1 driveway

뜻 입력 : 차고 진입로

새 단어가 단어장에 추가되었습니다.

3

# Java CRUD – Read

- 데이터 조회
  - WordManager class
    - 모든 단어 보기 메뉴 처리
  - WordCRUD class
    - `listAll()` 함수 생성
  - Word class
    - `toString()` : overriding

\*\*\* 영단어 마스터 \*\*\*

\*\*\*\*\*

1. 모든 단어 보기
2. 수준별 단어 보기
3. 단어 검색
4. 단어 추가
5. 단어 수정
6. 단어 삭제
7. 파일 저장
0. 나가기

\*\*\*\*\*

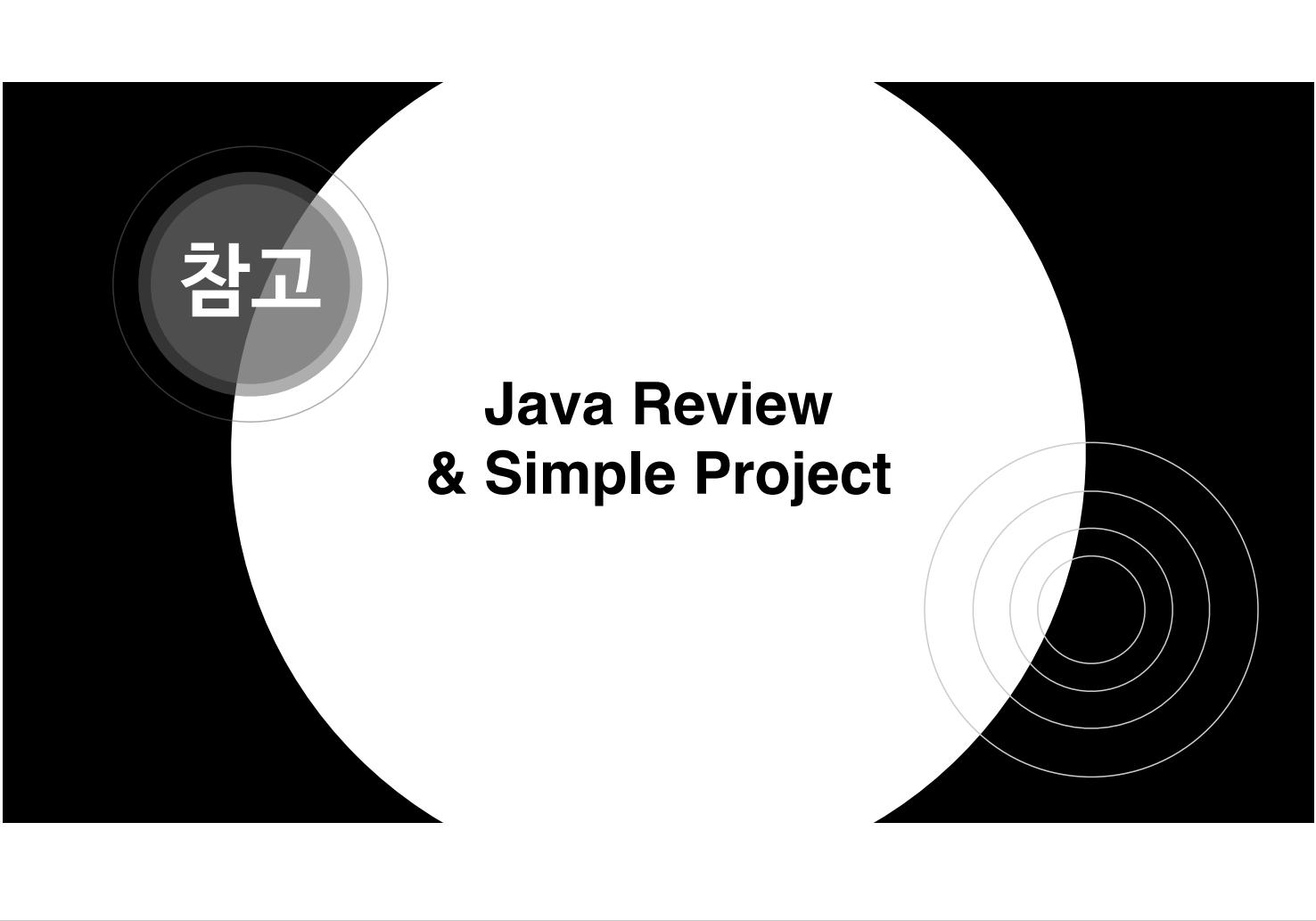
=> 원하는 메뉴는?

Read/ List

=> 원하는 메뉴는? 1

1 ***	superintendent	관리자, 감독관
2 *	electric	전기의, 전기를 생산하는
3 **	equipment	장비, 용품

4



참고

## Java Review & Simple Project

### Java 구성

- Java 기초 문법
  - 변수와 자료형 / 입출력
  - 산술, 논리, 관계, 대입 연산자
  - 조건문(if, switch) / 반복문(for, for-each, while, break, continue)
  - 배열(생성, 배열길이)
- 객체지향 프로그래밍(OOP)
  - 객체지향 개념 이해(SOLID 원칙)
  - 객체지향 프로그래밍 특징(상속, 다형성, 캡슐화, 추상화), overriding, overloading
  - 클래스 구성요소 (멤버변수, 멤버메소드, getter & setter)
  - class, interface, abstract class, inner class

# Java 구성

- Java 기본 라이브러리 (API)
  - 패키지, 클래스
  - java.util 패키지 / java.lang 패키지 등
  - Scanner, String, StringBuffer, Math, File, SimpleDateFormat, Wrapper class 등
- Java 파일 입출력
  - InputStream(byte 단위), Reader(문자 단위), BufferedReader(문자열 단위)
- Java Collection framework
  - List / Set / Map Interface
- Java 외부 라이브러리 연동
  - JDBC driver, RDBMS 종류마다 라이브러리 제공
  - 그외 다수

7

## Lab1 (1) 개발환경 설정 – Java IDE 설치

- Java 개발도구
  - OpenJDK or Oracle Java JDK  
<https://openjdk.java.net/>
- 통합개발환경 (IDE)
  - STS4(Spring Tool Suite) or Eclipse <https://spring.io/tools>
  - IntelliJ IDEA([free educational licenses](#))
- 버전관리
  - Git 설치 <https://git-scm.com/>
  - Github 가입 <https://github.com/>



8

The screenshot shows the official website for Spring Tools 4. At the top, there's a navigation bar with links for "Why Spring", "Learn", "Projects", "Training", "Support", and "Community". Below the navigation is a large green header with the "Spring Tools 4" logo. A sub-header text reads: "Spring Tools 4 is the next generation of Spring tooling for your favorite coding environment. Largely rebuilt from scratch, it provides world-class support for developing Spring-based enterprise applications, whether you prefer Eclipse, Visual Studio Code, or Theia IDE." Below this, there's a section titled "Spring Tools 4" with a sub-section "for Eclipse". It says "The all-new Spring Tool Suite 4. Free, Open source." There are three download links: "4.11.1 - LINUX X86\_64", "4.11.1 - MACOS X86\_64", and "4.11.1 - WINDOWS X86\_64". To the right of the text is a screenshot of the Eclipse IDE interface showing Java code and a debugger.

9

# Lab1 (1) 개발환경 설정 – OpenJDK 설치

- Oracle Java Downloads : <https://www.oracle.com/java/technologies/downloads>
- OpenJDK
  - The place to collaborate on an open-source implementation of the Java Platform, Standard Edition, and related projects
  - <https://openjdk.java.net>
- OpenJDK 배포판 설치 <https://aws.amazon.com/ko/corretto/> ✓
 

지금 Amazon Corretto 다운로드하기

[Amazon Corretto 8 다운로드하기](#)

[Amazon Corretto 11 다운로드하기](#)
- Amazon Corretto 11 다운로드 & 설치
- 환경변수 확인
  - JAVA\_HOME

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/amazon-corretto-11.jdk/Contents/Home
```

10

The screenshot shows the AWS Corretto 11 User Guide page. The left sidebar contains navigation links for 'Amazon Corretto' and 'Corretto 11 User Guide'. Under 'Downloads', there are links for 'What is Amazon Corretto 11?', 'List of Patches for Amazon Corretto 11', 'Linux', 'Windows' (selected), 'macOS', 'Docker', and 'Downloads' (selected). The main content area is titled 'Downloads for Amazon Corretto 11' and includes links for 'PDF', 'Kindle', and 'RSS'. A descriptive text explains that the topic lists all available downloads for Amazon Corretto 11, allowing users to choose between permanent URLs for the latest version or version-specific URLs. Below this is a section titled 'Amazon Corretto 11 Permanent URLs' with instructions on how to retrieve the latest Linux Corretto 11 .tgz package using a permanent URL via the CLI. A command-line example is shown: `wget https://corretto.aws/downloads/latest/amazon-corretto-11.x86_64.tgz`.

11

# Lab1 (1) 개발환경 설정 - git 설정

- Git 설치
  - <https://git-scm.com/>
- Git 초기 설정
  - 사용자 이름 설정

```
% git config --global user.name "yourname"
```



- 이메일 설정
 

```
% git config --global user.email "you@example.com"
```
- 확인
 

```
% git config --list
```

12

# Lab1 (2) Simple Project 제작

- STS4 or Eclipse or IntelliJ 실행
- Workspace 확인 : 프로젝트 저장공간 확인
- 새 프로젝트 생성
  - New > Java Project - **HelloWorld**
- 새 클래스 생성
  - New > Class - **Main**
    - 패키지 이름 입력 : **com.example.lab1**
    - public static void main(String [] args) 체크
- 코드 입력 후 실행 확인

```
Hello World!!!
이름 입력 : 홍길동
[홍길동]님 환영합니다~
```

13

The screenshot shows the Java Project creation process and its execution results.

**New Java Class Dialog:** A dialog box titled "New Java Class" is open. It shows the package name "com.example.lab1" selected. The class name is "Main". The "public" modifier is selected. The source folder is set to "HelloWorld/src". A message at the top says "Type already exists."

**Code Editor:** A code editor window titled "Main.java" displays the following Java code:

```
1 package com.example.lab1;
2
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         Scanner s = new Scanner(System.in);
8         System.out.println("Hello World!!!\n");
9         System.out.print("이름 입력 : ");
10        String name = s.nextLine();
11        System.out.println("[ " + name + " ]님 환영합니다~");
12    }
13 }
```

**Console Output:** The console output window shows the program's output:

```
Hello World!!!
이름 입력 : 홍길동
[홍길동]님 환영합니다~
```

14

# Lab1 (3) GitHub 연결 설정 (on GitHub)

- github 인증 방식 변경으로 인해 Access Token 생성 필요함
  - Id/password 인증대신 [id/Personal Access Token 방식](#)으로 변경 (2021년 8월)
- 설정 방법
  - 로그인 > Settings
  - Developer settings > Personal access token > Generate new token
    - Note 기입
    - Expiration 선택 - no expiration
    - Select scopes - repo 체크
  - 토큰 복사
- 새로운 원격저장소 생성(github) - 새 프로젝트 소스 버전관리
  - 저장소 URI 복사 > project에서 commit and push!

15

The screenshot shows the GitHub developer settings interface. On the left, there is a sidebar with navigation links: Signed in as kkim-hgu, Set status, Your profile, Your repositories, Your codespaces, Your projects, Your stars, Your gists, Feature preview, Help, Settings (which is selected), and Sign out. The main content area is titled 'Settings / Developer settings' and shows the 'Personal access tokens' section selected. The title 'New personal access token' is displayed. A note states: 'Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.' Below this is a 'Note' field containing 'JavaProjectLab'. A 'Expiration \*' field has 'No expiration' selected, with a note: 'The token will never expire!' A warning message in a blue box says: 'Beep bop! Tokens that live forever are scary. Expiration dates are highly recommended!' Under 'Select scopes', there is a checked checkbox for 'repo' with the description 'Full control of private repositories'. At the bottom, a message encourages users to copy their token: 'Make sure to copy your personal access token now. You won't be able to see it again!', followed by the token value 'ghp\_GgK1NJib9oFBhtUvxpTNvRvD3KrKhB00BZfA' and a copy icon.

16

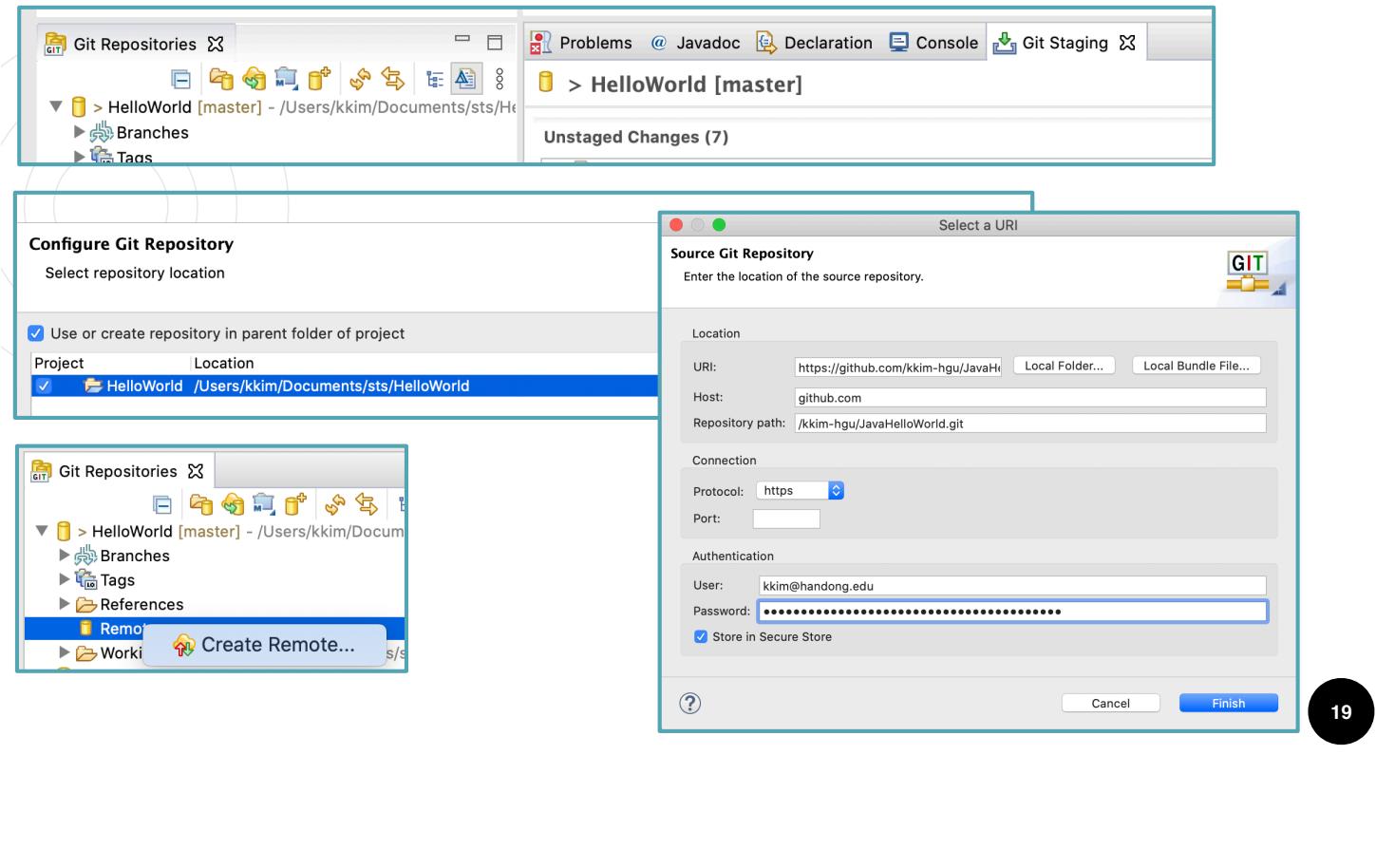
The screenshot shows the GitHub interface. On the left, a sidebar menu is visible with options like 'Your profile', 'Your repositories' (selected), 'Your codespaces', 'Your projects', 'Your stars', 'Your gists', 'Feature preview', 'Help', 'Settings', and 'Sign out'. The main area shows the 'Create a new repository' form. It has fields for 'Owner' (set to 'kkim-hgu') and 'Repository name' (set to 'JavaHelloWorld'). Below these, there's a note about repository names and a 'Description (optional)' field. Under 'Visibility', 'Public' is selected. At the bottom of the form, there's a note about anyone on the internet being able to see the repository. To the right, the repository details for 'kkim-hgu / JavaHelloWorld' are shown, including tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and a star count of 1, fork count of 0, and a link to the repository at <https://github.com/kkim-hgu/JavaHelloWorld.git>. A 'Quick setup' section provides instructions for setting up the repository on desktop or via HTTPS/SSH.

17

## Lab1 (3) GitHub 연결 설정 (on sts4)

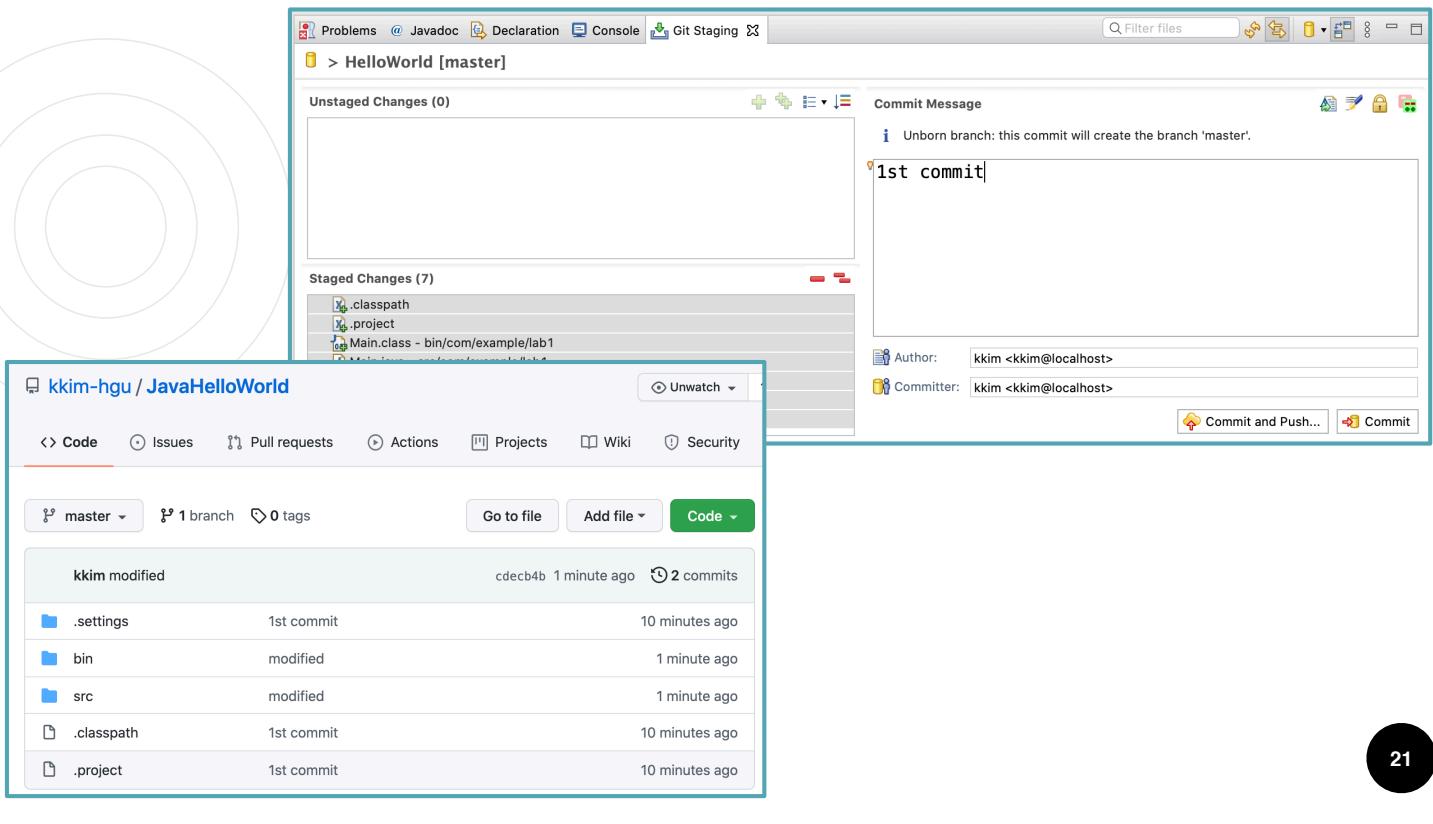
- 버전 관리에 필요한 화면 구성 설정
  - Window View 추가 : Window > Show View > Git Staging, Git Repositories
- 로컬 저장소 생성
  - 프로젝트 우클릭 > Team > Share Project
    - Use or existing repository or create a new one 체크
    - Create Repository 클릭 > finish
- 원격 저장소 연결(Remote Repository)
  - Git Repositories > [프로젝트] > Remotes 우클릭 > Create Remote
  - Source Git Repository의 URI, 인증 설정
  - Git Repositories > [프로젝트] > Remote > origin > Change에서 저장소 URI 붙여넣기
  - Id/ Personal Access token, Store in Secure store 체크, id 넣고 pw에 토큰 붙여넣기

18



## Lab1 (4) 프로젝트 소스 push (on sts4)

- **소스 파일 commit & push**
  - 프로젝트 우클릭 > Team > Commit 클릭
  - Git Staging 창 > Unstaged Changes 항목 전체 선택하여 [++] 클릭
  - Commit Message 기입 후 Commit and Push
- **on GitHub**
  - 저장소에 정상적으로 올려졌는지 확인
- **소스코드 업데이트**
  - 소스파일 수정 후 저장
  - Commit and Push
  - GitHub 확인 - commit 횟수 확인!



21

# Build Tool

- **Build**

- 코드와 의존성 라이브러리를 이용하여 컴파일하고 패키지로 만드는 작업

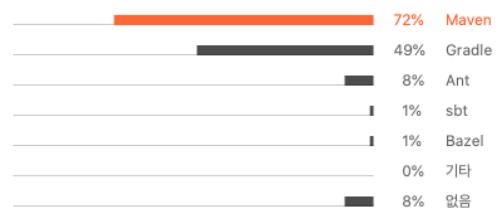
- **Build Tool**

- 프로젝트 생성, 컴파일, 테스트, 패키징, 배포, 문서화 등 작업을 지원하는 통합 프로그램
- 라이브러리 관리 및 버전 동기화

- **종류**

- Ant, Maven, Gradle 등
- 2021년 기준 Maven 비율이 가장 높음
- 신규 프로젝트는 Gradle 이 많아지는 추세

어떤 빌드 시스템을 주로 사용하시나요?



<https://www.jetbrains.com/ko-kr/lp/devcosystem-2021/java/>

22

# Build Tool – Gradle

- Groovy를 이용한 빌드 자동화 시스템
  - Java, java web, C, C++, Android 등의 여러 가지 언어 지원
  - 현재 Android APP을 만들 때 사용하는 Android Studio 공식 빌드 시스템
- 특징
  - 빌드 스크립트를 XML 언어가 아닌 JVM에서 동작하는 스크립터 언어 ‘Groovy’ 기반의 DSL(Domain Specific Language) 사용
  - 자바문법과 유사하며 쉽게 익힐 수 있음
  - Ant와 Maven 등 이전 빌드 도구의 단점 보완 및 장점을 취합한 오픈소스 공개 빌드 도구
  - 의존성 관리를 위한 다양한 방법을 제공
- build.gradle 파일 이용
  - 빌드 구성 스크립트(Build Configuration Script)를 통해 의존성, 플러그인 등 설정

<https://ko.wikipedia.org/wiki/Gradle>

23

# Build Tool – Maven

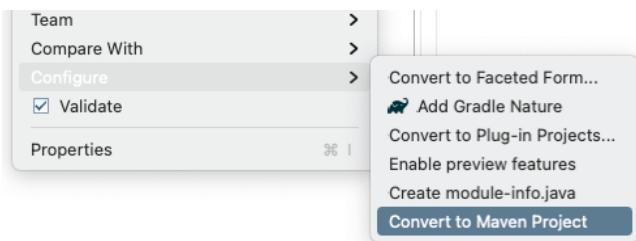
- 특징
  - java 프로젝트 관리도구
  - Apache Ant 대안으로 만들어짐
  - 전체적인 라이프 사이클을 관리하며, 핵심 라이브러리(Jar)와 의존 라이브러리 관리 용이
  - Maven 환경설정 파일 : **pom.xml**
  - pom.xml에 정의한 라이브러리를 자동으로 다운로드하여 설정해 줌
  - 간단한 설정을 통한 배포 관리 가능

<https://www.jetbrains.com/ko-kr/lp/devcosystem-2021/java/>

24

# Maven Project로 만들기

- 프로젝트 생성, 제작 후
- 프로젝트 우클릭 → Configure → Convert to Maven Project 클릭
- pom.xml 파일이 자동 생성됨
- 프로젝트 루트 폴더에 위치
- 이 파일을 통해 필요한 설정 추가/변경
- 프로젝트 우클릭 Maven 메뉴 표시됨



```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>Lab2Maven</groupId>
  <artifactId>Lab2Maven</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <release>10</release>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

25

# References

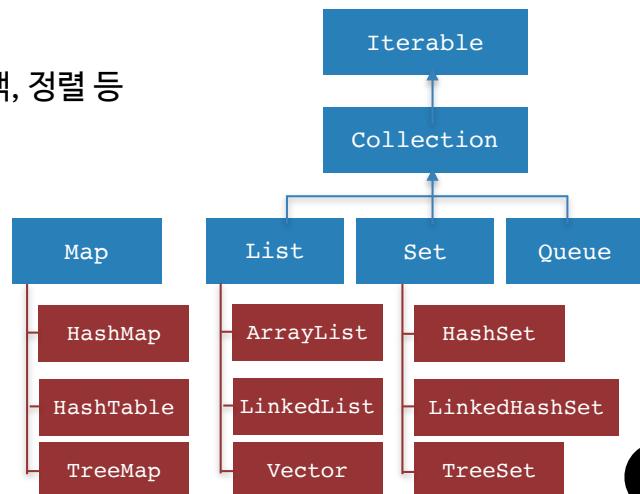
- Java 문법 참고 사이트
  - <https://www.w3schools.com/java/default.asp>
  - <http://tcpschool.com/java/intro>
  - JDK APIs Documentation
    - <https://docs.oracle.com/en/java/javase/16/docs/api/index.html>
- 코딩연습사이트
  - 백준 <https://www.acmicpc.net/>
  - 프로그래머스 [https://programmers.co.kr/learn/challenges?tab=all\\_challenges](https://programmers.co.kr/learn/challenges?tab=all_challenges)
  - LeetCode <https://leetcode.com/problemset/all/?page=1>

26

# Java collection framework

- 다수의 데이터를 쉽고 효과적으로 처리할 수 있도록 제공하는 클래스의 집합
  - 데이터를 저장하는 자료 구조
  - 데이터 처리를 위한 알고리즘 구조
  - 동적 자료 관리가 가능 : 추가, 수정, 삭제, 검색, 정렬 등
- Interface 제공
  - List interface : 순서 보장, 중복허용
  - Set interface : 순서 보장되지 않고, 중복 허용하지 않음
  - Map interface : key와 value로 이루어짐

<https://www.javatpoint.com/collections-in-java>



27

## Methods of Collection Interface

- **public boolean add(E e)**
- **public boolean addAll(Collection<? extends E> c)**
- **public boolean remove(Object element) / public boolean removeAll(Collection<?> c)**
- **public int size()**
- **public void clear()**
- **public boolean contains(Object element)**
- **public Iterator iterator()**
- **public Object[] toArray() / public <T> T[] toArray(T[] a)**
- **public boolean isEmpty()**
- **public boolean equals(Object element)**
- **...**

28

# ArrayList class

```
Module java.base
Package java.util
Class ArrayList<E>
java.lang.Object
    java.util.AbstractCollection<E>
        java.util.AbstractList<E>
            java.util.ArrayList<E>
```

- 순서대로 저장되며 중복 저장이 가능한 클래스
- 주요 함수
  - public boolean add(E e)
  - public E get(int index)
  - public boolean remove(Object element)
  - public boolean removeAll(Collection<?> c)
  - public int size()
  - public void clear()
  - public Object clone()
  - public boolean contains(Object element)
  - public boolean equals(Object element) ...

29

# ArrayList 연습 (in class)

- Student class 생성
  - 멤버변수 : 번호, 이름, 나이
  - 생성자, Getters & Setters 생성
  - Overriding toString() 함수 만들기
- Main - main()
  - ArrayList<Student> 생성
  - 학생 3명 정보 추가
  - 학생 리스트 출력
  - 학생 1명 정보 제거
  - 학생 리스트 출력

```
1 package com.example.lab2;
2
3 public class Student {
4     int no;                      // number
5     String name;                 // name
6     int age;                     // age
```

```
Student [no=101, name=Kim, age=23]
Student [no=102, name=Lee, age=21]
Student [no=103, name=Park, age=25]
One student removed!
Student [no=101, name=Kim, age=23]
Student [no=103, name=Park, age=25]
```

```

1 package com.example.lab2;
2 import java.util.List;
3 import java.util.ArrayList;
4
5 public class Main {
6     public static void main(String args[]){
7         //Create arraylist of user-defined class objects
8         List<Student> al=new ArrayList<Student>();
9
10        Student s1=new Student(101,"Kim",23);
11        al.add(s1);//adding Student class object
12        Student s2=new Student(102,"Lee",21);
13        al.add(s2);//adding Student class object
14        al.add(new Student(103,"Park",25));
15
16        //Traverse elements of arraylist (method #1)
17        for (Student s : al){
18            System.out.println(s.toString());
19        }
20
21        //Remove an object from arraylist
22        al.remove(s2);
23        System.out.println("One student removed!");
24
25        //Traverse elements of arraylist (method #2)
26        for (int i=0; i<al.size(); i++){
27            System.out.println(al.get(i).toString());
28        }
29    }
30 }
```

```

1 package com.example.lab2;
2
3 public class Student {
4     int no;          // number
5     String name;    // name
6     int age;         // age
7
8     public Student(int no, String name, int age) {
9
10    }
11
12    public int getNo() {
13        return no;
14    }
15    public void setNo(int no) {
16        this.no = no;
17    }
18    public String getName() {
19        return name;
20    }
21    public void setName(String name) {
22        this.name = name;
23    }
24    public int getAge() {
25        return age;
26    }
27    public void setAge(int age) {
28        this.age = age;
29    }
30
31    @Override
32    public String toString() {
33        return "Student{" +
34                "no=" + no +
35                ", name='" + name + '\'' +
36                ", age=" + age +
37                '}';
38
39 }
40 }
```

**Student.java**

**Main.java**