

תוכנה 1 בשפת Java

שיעור מספר 1: "שלום עולם"

פרופ' ליאור וולף

בית הספר למדעי המחשב
אוניברסיטת תל אביב



presented by kenichi-naitou.

מה בתכנית?

- טעימה משפת Java
- פונקציית `main`
- 8 הטיפוסים היסודיים
- ביטויים ואופרטורים
- טיפוס המחרוזת וטיפוס המערך



שלום עולם

```
C:\WINDOWS\system32\cmd.exe
C:\>
C:\>javac HelloWorld.java
```

```
HelloWorld.java - Notepad
File Edit Format View Help
public class HelloWorld {
    public static void main(String[] arguments) {
        System.out.println("Hello world");
    }
}
```

HelloWorld.java

compile



HelloWorld.class

Run on Windows

Run on Solaris

Run on Mac

```
C:\WINDOWS\system32\cmd.exe
C:\>
C:\>java HelloWorld
```



SOLARIS



Write Once – Run Anywhere !

המפרש (interpreter)

■ את הקוד שנכתב בשפת Java מריץ מפרש
■ בדומה לשפת Python

■ לריצה בעזרת מפרש יש כמה חסרונות:

■ מאט את מהירות הריצה

■ טעויות מתגלות רק בזמן הריצה

■ לצורך כך הוסיפו ב Java שלב נוסף – *הידור*
(compilation)

המהדר (compiler)

- מבצע עיבוד מקדים של קוד התוכנית (שכתובה בקובץ טקסט רגיל) ויוצר קובץ חדש בפורמט **נוח יותר**
- קובץ זה אינו קריא למתכנת אנושי (אף שניתן לפתוח אותו בעורך טקסט כגון Notepad), אולם המבנה שלו מותאם לקריאה ע"י המפרש של Java
- פורמט זה נקרא byte code והוא נשמר בקובץ עם סיומת .class
- בתהליך העיבוד ("קומפילציה") נבדק התחביר של הקוד – והשגיאות המתגלות מדווחות למתכנת

יבילות (portability)

- מדוע אנו מסתפקים בפורמט "נוח יותר"?
- מדוע אין המהדר יוצר קובץ בפורמט התואם בדיוק לחומרת המחשב, וכך היה נחסך בזמן ריצה גם שלב ה"הבנה" של הקוד?
- זאת מכיוון שאיננו יודעים מראש על איזה מחשב בדיוק תרוץ תוכנית ה-Java שכתבנו
- תוכניות Java חוצות סביבות (cross platform)
- סביבה = חומרה + מערכת הפעלה
- תוכנית שנכתבה והודרה במחשב מסוים, תוכל לרוץ בכל מחשב אשר מותקן בו מפרש ל-Java

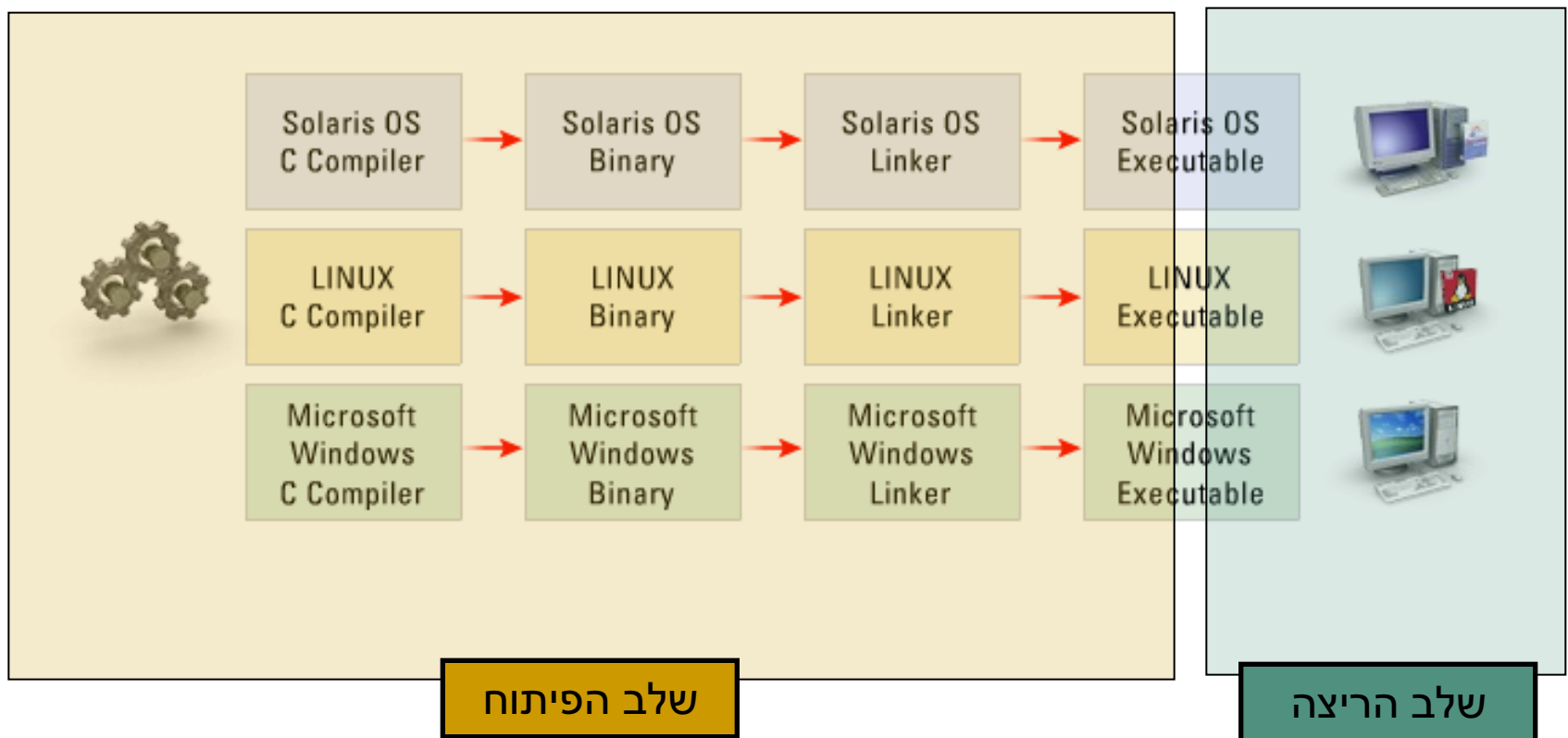
המכונה המדומה

(Java Virtual Machine)

- הקובץ המהודר מכיל הוראות ריצה ב"מחשב כללי" – הוא אינו עושה הנחות על ארכיטקטורת המעבד, מערכת ההפעלה, הזיכרון וכו'...
- עבור כל סביבה (פלטפורמה) נכתב מפרש מיוחד שיודע לבצע את התרגום מהמחשב הכללי, המדומה, למחשב המסוים שעליו מתבצעת הריצה
- את המפרש לא כותב המתכנת!
- דבר זה כבר נעשה ע"י ספקי תוכנה שזה תפקידם, עבור רוב סביבות הריצה הנפוצות

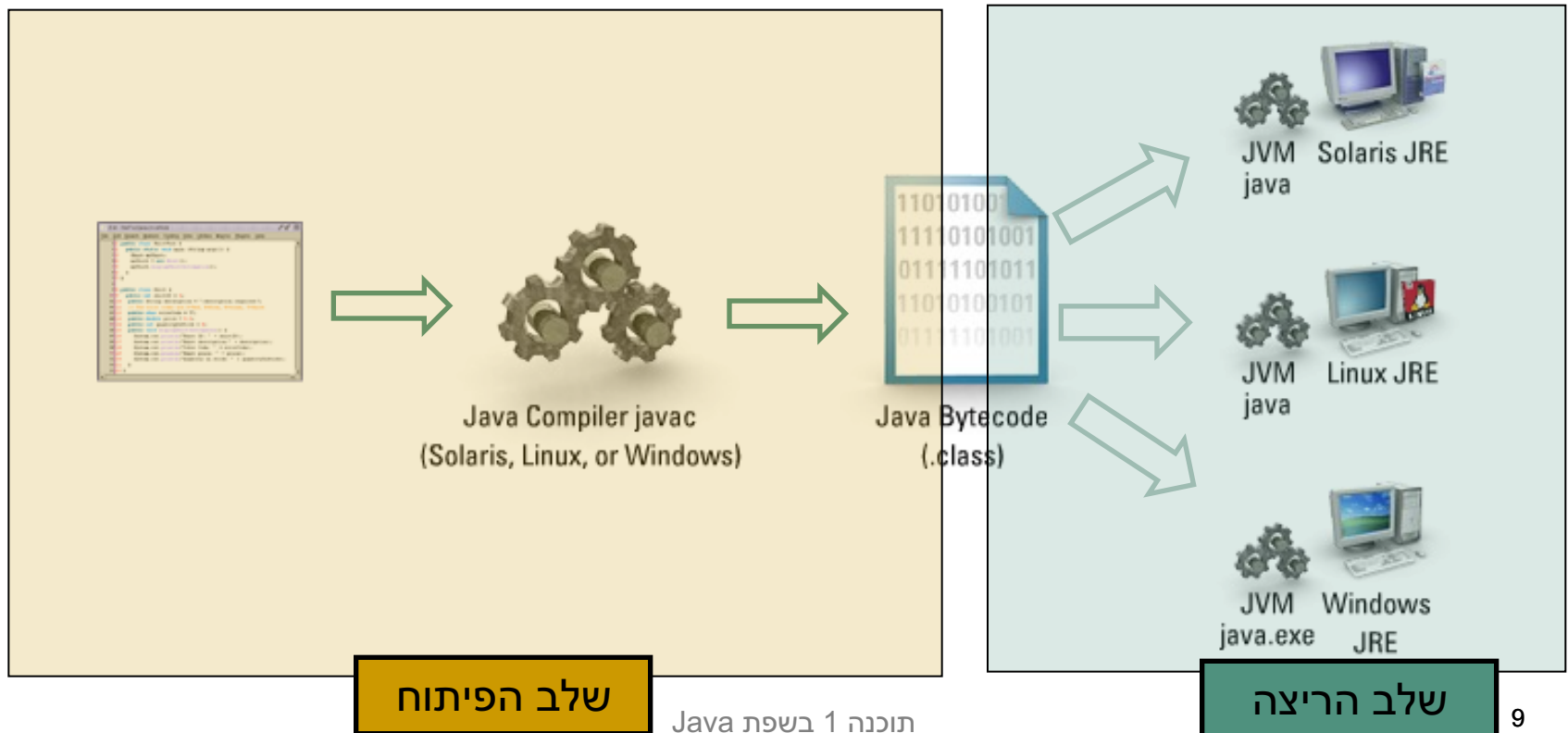
תלות בסביבה (platform specific)

■ בשפות אחרות (C/C++) אין הדבר כך:



עצמאות סביבתית (platform independence)

■ ב Java תכונה זו אפשרית הודות לרעיון "שפת הביניים" וה JVM הנפרד לכל סביבה



תוכנה 1 בשפת Java
אוניברסיטת תל אביב



שלום עולם

הגדרת מחלקה בשם HelloWorld.
בשלב זה, נזהה מחלקה עם קובץ באותו שם

המחלקה ציבורית –
ניתן להשתמש בה ללא
הרשאות מיוחדות

חתימת המתודה

```
public class HelloWorld {
```

```
public static void main(String[] arguments) {
```

```
System.out.println("Hello World");
```

```
}
```

גוף המתודה

הגדרת מתודה (פונקציה)

יצירת תחום (scope)

המתודה main

```
public static void main(String[] arguments) {  
    System.out.println("Hello World");  
}
```

- כאשר אנו מריצים מחלקה ה JVM מחפש מתודה עם חתימה זו, ומריץ אותה
 - main – שם המתודה
 - public - המתודה ציבורית – ניתן להשתמש בה ללא הרשאות מיוחדות
 - static – מתודה של המחלקה (יוסבר בהמשך)
 - void – טיפוס הערך המוחזר. למתודה זו אין ערך מוחזר (ריק = void)

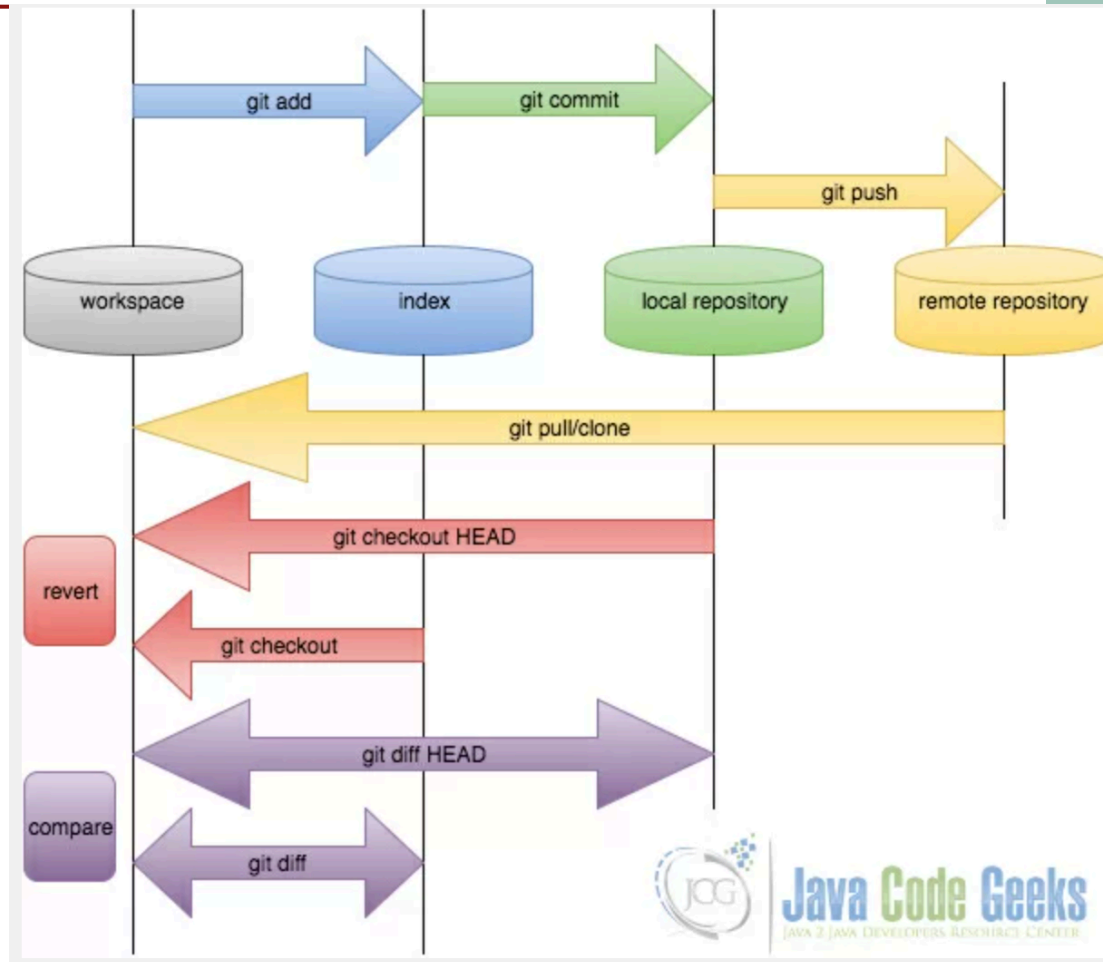
Git

בית הספר למדעי המחשב
אוניברסיטת תל אביב

מה מאפשר לנו ה Git

- ניהול גירסאות של הקוד:
 - מעקב אחרי שינויים.
 - חזרה אחורה.
 - גיבוי.
- עבודה בצוות:
 - עדכון של שינויים שבוצעו ע"י חברי צוות אחרים.
 - מיזוג (אוטומטי או ידני) במידה ושני חברי צוות משנים את אותו הקוד.
- ככה עובדים היום בכל מקום (ואם זה לא Git, זה כלי בקרת תצורה אחר עם מאפיינים דומים)

איך זה נראה?



<https://examples.javacodegeeks.com/software-development/git/git-tutorial-beginners/> מתוך:

תוכנה 1 בשפת Java
אוניברסיטת תל אביב

מה עושים בשביל להתחיל?

- שלב ראשון: פותחים חשבון ב [GitHub](#) (שיכיל את ה remote repositories שלנו).
- ניתן לפתוח חשבון חינמי עם כל כתובת מייל, אבל חשובן זה מאפשר לכם לפתוח repositories פומביים בלבד.
- הרשמה עם כתובת המייל האוניברסיטאית תאפשר לכם לפתוח repositories עם ניראות פומבית או פרטית.
- ניתן כמובן לפתוח יותר מחשבון github אחד.
- בשביל הקורס עליכם לפתוח חשבון עם שם המשתמש האוניברסיטאי שלכם.

מה עושים בשביל להתחיל?

- שלב שני: התקנת git על המחשב שלכם. ניתן להעזר במדריך הבא:

<https://github.com/git-guides/install-git>

- ניתן לעבוד ב git בכמה אופנים:
 - Command line (גם ב windows ניתן לעבוד ב command line בצורה פשוטה – התקנת ה git מכילה גם git bash – כלי שמאפשר עבודה מהטרמינל).
 - מנשק משתמש גרפי.
 - שילוב של Git בתוך הכלי הפיתוח שלכם (Eclipse/IntelliJ).

מה עושים בשביל להתחיל?

■ שלב שלישי: החליטו מה אתם רוצים לעשות – לפתוח פרוייקט חדש או לשכפל פרוייקט קיים.

שכפול פרוייקט קיים

<https://github.com/lenadank/Software1GitDemo>

כשנכנסים לתיקיה היא נראית כך:

lenadank / Software1GitDemo

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

| | | | | |
|-------|--------------|---------|--------------|-----------|
| lenad | first commit | 69d5803 | 15 hours ago | 1 commits |
| src | first commit | | 15 hours ago | |

נריץ את הפקודה הבאה:

```
git clone https://github.com/lenadank/Software1GitDemo.git
```

נפתח את הפרוייקט דרך ה eclipse באופן הבא: ניצור פרוייקט חדש ע"י
.file->new project

יצירת פרוייקט חדש

- צרו פרוייקט חדש וריק ב `github`.
- עקבו אחרי ההנחיות.

פיילוט עבור הגשת תרגילי בית

■ היכנסו לקישור הבא:

<https://classroom.github.com/a/-SLTSG93>

■ עליכם להתחבר על github עם משתמש ה git שמחובר לחשבון האוניברסיטאי שלכם (בדוגמא המצורפת התחברנו עם המשתמש software1forever).

You're ready to go!

You accepted the assignment, **hw0_pilot**.

First, accept your invitation to software1course2021a:

✉ <https://github.com/orgs/software1course2021a/invitation>

Then, access your assignment:

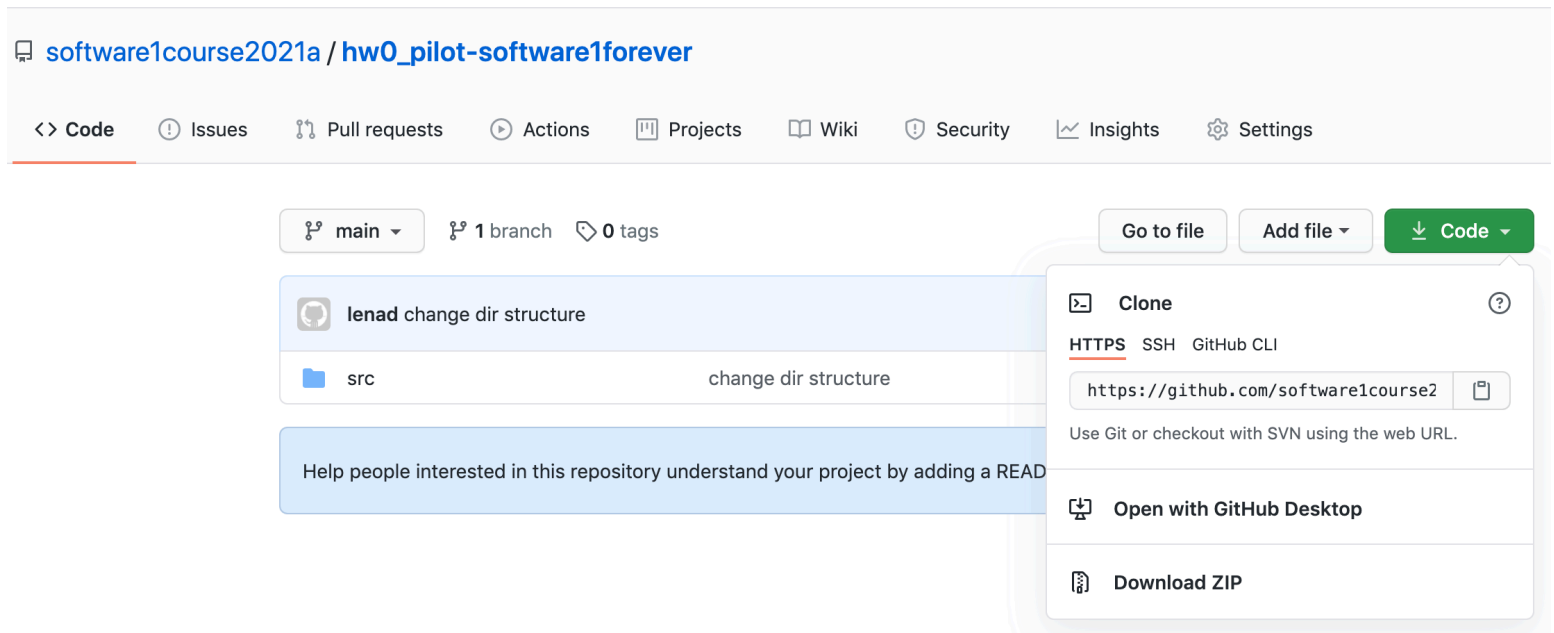
📄 https://github.com/software1course2021a/hw0_pilot-software1forever

We've configured the repository associated with this assignment ([update](#)).

לחצו כאן!

פיילוט עבור הגשת תרגילי בית

- הפרוייקט יפתח בתצורה המוכרת של פרוייקט קיים.
- לחצי עם הכפתור הירוק code והעתיקו את הקישור המצורף (עדיף לבחור ב https)



The screenshot shows a GitHub repository page for `software1course2021a / hw0_pilot-software1forever`. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there are buttons for 'main', '1 branch', and '0 tags'. A commit by 'lenad' is visible with the message 'change dir structure' and a 'src' folder. A 'Code' button is highlighted, and a dropdown menu is open, showing options to 'Clone' (with sub-options for HTTPS, SSH, and GitHub CLI), 'Open with GitHub Desktop', and 'Download ZIP'. The HTTPS URL is `https://github.com/software1course2`.