

תוכנה 1

תרגול 1: מנהלות, סביבת העבודה ומבוא ל-
Java

בית הספר למדעי המחשב
אוניברסיטת תל אביב

מנהלות |

• מתרגלים:

- יעל אמסטרדמר (שעת קבלה: חמישי ב-12, בתיאום מראש)
- דביר נתנאלי (שעת קבלה: רביעי ב-19, בתיאום מראש)

• אתר הקורס:

<http://courses.cs.tau.ac.il/software1/1415b>

• אתר הקורס ב-moodle

- הגשת תרגילים
- פורום הקורס (שאלות ותשובות על תרגילי הבית ובכלל)
- הודעות המרצה

✓ מי שאינו רשום לקורס באתר Moodle, שישלח מייל למתרגל עם שם משתמש אוניברסיטאי, שם מלא ות"ז.

מנהלות II

• תרגילי בית

- יינתנו 10 תרגילים המהווים 15 נקודות מהציון הסופי בקורס (כל תרגיל יהיה במשקל 1 או 2 נקודות).
- הגשה בימי חמישי אחת לשבוע או שבועיים, דרך מערכת המודל.
- יש להגיש לפחות 6 תרגילים, ולצבור לפחות 9 נקודות כדי לגשת למבחן.
- מבחינת הציון הסופי, אין חשיבות לציון התרגילים כל זמן שעברתם.
- לזכותכם 5 "ימי חסד" המאפשרים הגשה באיחור ללא בקשת אישור.
- חשוב להגיש את התרגילים בצורה מדויקת ובלי להעתיק.
 - מימוש מדויק על פי ההוראות בתרגיל
 - להקפיד על פלטים מדויקים כי הבדיקה עשויה להיות אוטומטית
 - הגשת קובץ zip עם שם נכון הכולל קבצי קוד עם סיומת java ולא קבצי class
- הגשת תרגיל מספר 1 ביום חמישי הבא !

מנהלות III

- עבודה במעבדות האוניברסיטה
- סביבת העבודה היא Linux.
- נדרש חשבון אישי במחשבי האוניברסיטה.
- הנחיות לפתיחת חשבון והכרת סביבת העבודה באתר הקורס.
- הסבר על הפעלת אקליפס במחשבי האוניברסיטה:
<http://courses.cs.tau.ac.il/software1/1415b/misc/lab-eclipse.pdf>
- עבודה מהמחשב הביתי
- יש להתקין Java 8 ו-Eclipse.
- הסבר מפורט לגבי השמשת סביבת העבודה מהבית:
<http://courses.cs.tau.ac.il/software1/1415b/misc/workenv.pdf>

חבילות הפיתוח וההרצה של Java

- גרסת ה-Java שעמה נעבוד:

Java SE (Standard Edition) 8.0

- חבילת סביבת ההרצה (נדרשת להרצת תוכניות ג'אווה):

JRE (Java Runtime Environment) that includes:

- JVM (Java Virtual Machine)
- Standard Class Library

- חבילת ערכת הפיתוח (נדרשת לפיתוח תוכניות ג'אווה):

JDK (Java Development Kit) that includes:

- JRE
- Command line tools: compiler, debugger etc.

- הורדה ותיעוד ב-

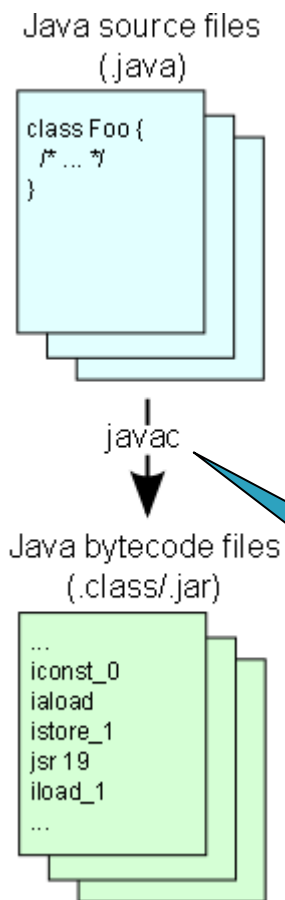
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

תהליך הפיתוח וההרצה של תוכנית ג'אווה

- ישנם שני שלבים נפרדים:

1. שלב פיתוח התוכנית

- בשלב זה אנו משתמשים במהדר (Compiler) כדי להמיר קבצי java. (קבצי טקסט הקריאים למתכנת) לקבצי class. שנועדו עבור המפרש (אינטרפרטר).
- המהדר מוודא את תקינות הקוד ומציג הודעות קומפילציה במקרה הצורך.
- הקומפיילר ניתן להרצה משורת הפקודה ע"י javac.



קומפילציה

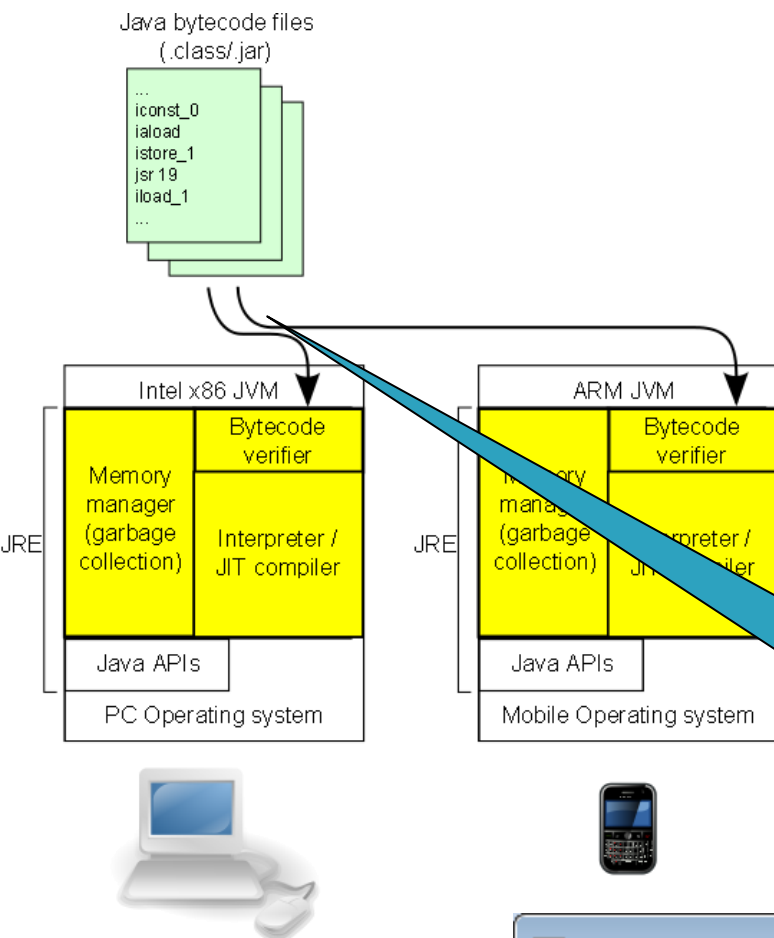
```
C:\Windows\system32\cmd.exe
C:\java>javac MyClass.java
```

תהליך הפיתוח וההרצה של תוכנית ג'אווה

• ישנם שני שלבים נפרדים:

2. שלב הרצת התוכנית

- בשלב זה אנו משתמשים במפרש (Interpreter) כדי להריץ את קבצי ה-class שיצרנו.
- ב-Java אותו קובץ class יכול לרוץ בסביבות שונות אם קיים עבורן מפרש.
- קבצי ה-class כתובים בפורמט bytecode.
- ניתן להריץ קובץ class משורת הפקודה ע"י הרצת הפקודה java.

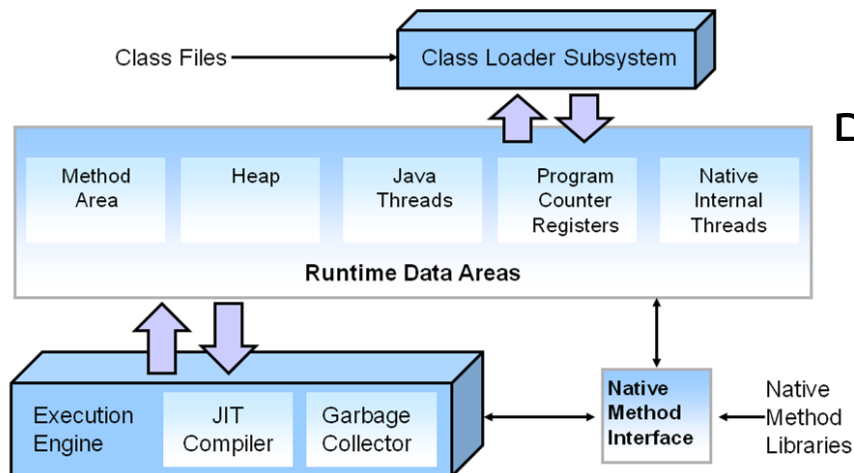


```
C:\Windows\system32\cmd.exe
C:\java>java MyClass
```

Java Virtual Machine

- ה-JVM היא "מכונה וירטואלית" המפרשת ומריצה Java ByteCode
- משמשת כשכבת תיווך בין מערכת ההפעלה לבין תוכניות הרצות על גביה.

HotSpot JVM: Architecture



- יודעת לטעון תוכניות
- יודעת לוודא את תקינות הקבצים הנטענים
- מכילה את המפרש (Interpreter)

- השימוש במכונה וירטואלית הוא ממאפייני היסוד של פלטפורמת Java והוא מאפשר לתוכניות שנכתבו בשפת Java ועברו הידור באמצעות המהדר של השפה, לרוץ על מגוון רחב של פלטפורמות חומרה ומערכות הפעלה.

IDE: סביבת פיתוח שלובה

IDE = Integrated Development Environment •

• סביבה המשלבת רכיבי/כלי פיתוח עצמאיים:

• עורך טקסט (editor)

• סייר הקבצים (browser)

• מהדר (compiler)

• סביבת זמן ריצה (JRE)

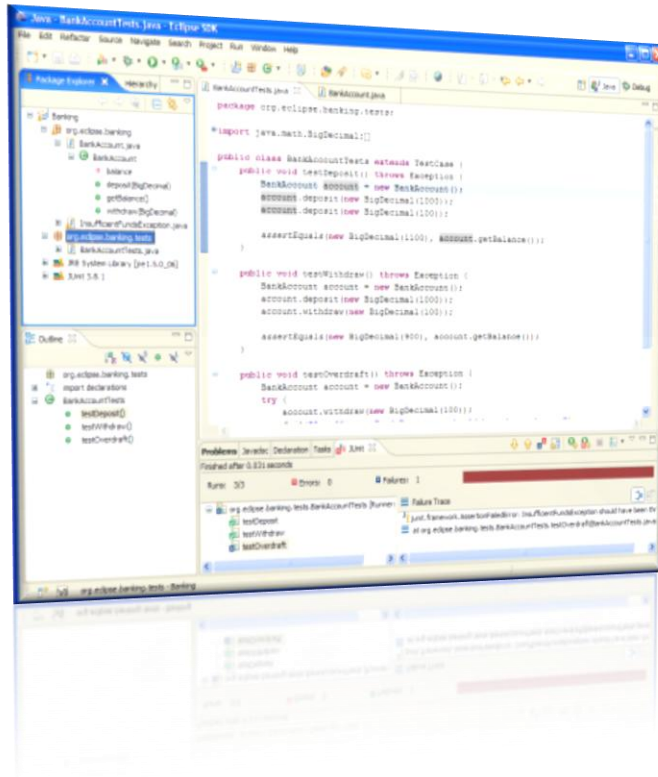
• מנפה השגיאות (debugger)

• ועוד...

• **Eclipse** – ה-IDE בו נשתמש בקורס.

• ניתן לערוך, לקמפל ולהריץ תוכניות ג'אווה מתוך אקליפס בצורה נוחה.

Eclipse



- IDE מתקדם המתאים גם לפיתוח תוכנה ב-Java.
- ניתן להתקנה ב-Linux, Windows ועוד...
- דורש התקנה בנפרד של JRE (או JDK)
- אתר הבית: www.eclipse.org מכיל אוסף גדול של מאמרים.
- הכרות:
- [דפי עבודה ללימוד Eclipse](#) באתר הקורס
- מדריך מומלץ לאקליפס: Vogella.com

Data Types

טיפוסי השפה

- **טיפוסים יסודיים (פרימיטיביים):** 8 טיפוסים מוגדרים בשפה שמיועדים להכיל ערכים פשוטים:
 - מספרים שלמים: **byte, short, int, long**
 - מספרים ממשיים: **float, double**
 - תווים: **char**
 - ערכים בוליאניים: **boolean**
- **טיפוסי הפנייה (References):** טיפוסים מורכבים היכולים גם להכיל מידע וגם לספק שירותים (יוסבר בהמשך)
 - המתכנת יכול להגדיר טיפוסי הפנייה חדשים
 - דוגמאות מיוחדות: מחרוזות ומערכים



הטיפוסים הפרימיטיביים

- בזיכרון המחשב נשמר המידע בפורמט בינארי
- **סיבית** (bit): ספרה בינארית ('0' או '1')
- **בייט** (octet, byte): קבוצה של 8 סיביות

	Type	Size	Value Range
שלמים	long	64 bits	$-2^{63} \rightarrow 2^{63}-1$
	int	32 bits	$-2^{31} \rightarrow 2^{31}-1$
	short	16 bits	$-2^{15} \rightarrow 2^{15}-1$
	byte	8 bits	$-128 \rightarrow 127$
ממשיים	double	64 bits	beyond the scope of the discussion
	float	32 bits	
תווים	char	16 bits	most alphabet languages
ערכים לוגיים	boolean	"1 bit"	true, false

הטיפוסים הפרימיטיביים

Primitive Types					
Type Name	Wrapper class	Value	Range	Size	Default Value
byte	java.lang.Byte	integer	-128 through +127	8-bit (1-byte)	0
short	java.lang.Short	integer	-32,768 through +32,767	16-bit (2-byte)	0
int	java.lang.Integer	integer	-2,147,483,648 through +2,147,483,647	32-bit (4-byte)	0
long	java.lang.Long	integer	-9,223,372,036,854,775,808 through +9,223,372,036,854,775,807	64-bit (8-byte)	0
float	java.lang.Float	floating point number	$\pm 1.401298E-45$ through $\pm 3.402823E+38$	32-bit (4-byte)	0.0
double	java.lang.Double	floating point number	$\pm 4.94065645841246E-324$ through $\pm 1.79769313486232E+308$	64-bit (8-byte)	0.0
boolean	java.lang.Boolean	Boolean	true or false	8-bit (1-byte)	false
char	java.lang.Character	UTF-16 code unit (BMP character or a part of a surrogate pair)	'\u0000' through '\uFFFF'	16-bit (2-byte)	'\u0000'

ASCII Table – char values

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

ערכי ברירת מחדל

Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	false
String (or any object)	null

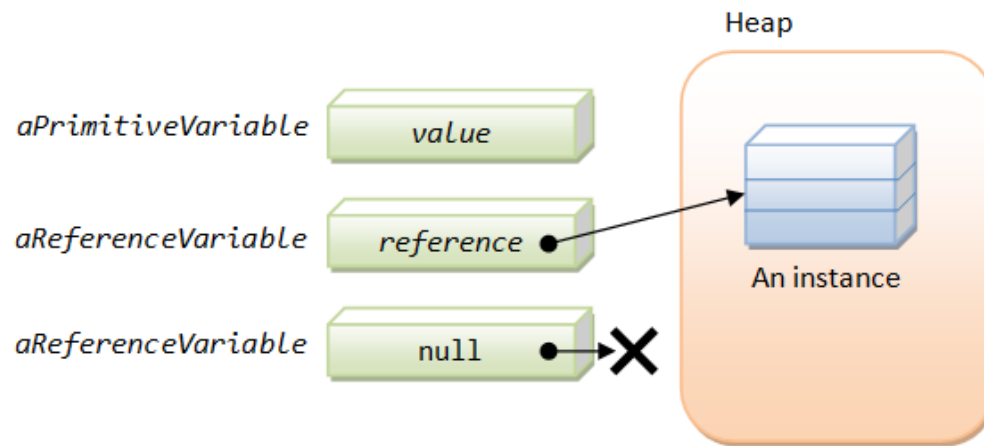
Reference Types

טיפוסים לא פרימיטיביים

- ג'אווה תומכת גם בטיפוסים מורכבים שאינם פרימיטיביים.
- טיפוסים אלו מוגדרים ע"י מחלקה שבתוכה נתונים ופונקציות רלבנטיות.
- מכל טיפוס ניתן ליצור מופעים רבים (אובייקטים).
- מקביל ליצירת משתנים רבים מטיפוס פרימיטיבי מסויים.
- אנו כמתכנתים נוכל להשתמש בקוד שלנו ב-
- אלפי הטיפוסים המוגדרים בספריה התקנית של Java, ה-Java API (Application Programming Interface)
<http://docs.oracle.com/javase/8/docs/api/>
- נוכל גם להגדיר מחלקות שייצגו טיפוסים לא פרימיטיביים נוספים על פי צרכינו.

טיפוסים לא פרימיטיביים

- טיפוסי הפניה (references) הם משתנים שמצביעים אל אובייקטים.



- דוגמא: הגדרת מחרוזת (String):

```
String myString = new String("Hello World");
```

Non-primitive type

Almost the same as:

```
String myString = "Hello World";
```

עצם (object)

מחרוזות

- מחרוזות הן אובייקט המחזיק אוסף של תווים.
- אופרטור שרשור:
- "Hello " + "World" is "Hello World"
- "19" + 8 + 9 is "1989"
- דוגמאות לפונקציות מהמחלקה String:

```
String str1 = "Hello";
char c = str1.charAt(0); // c == 'H'
String str2 = str1.toUpperCase(); // str2 == "HELLO"
int strLength = str1.length(); // i == 5
```

עוד ב-

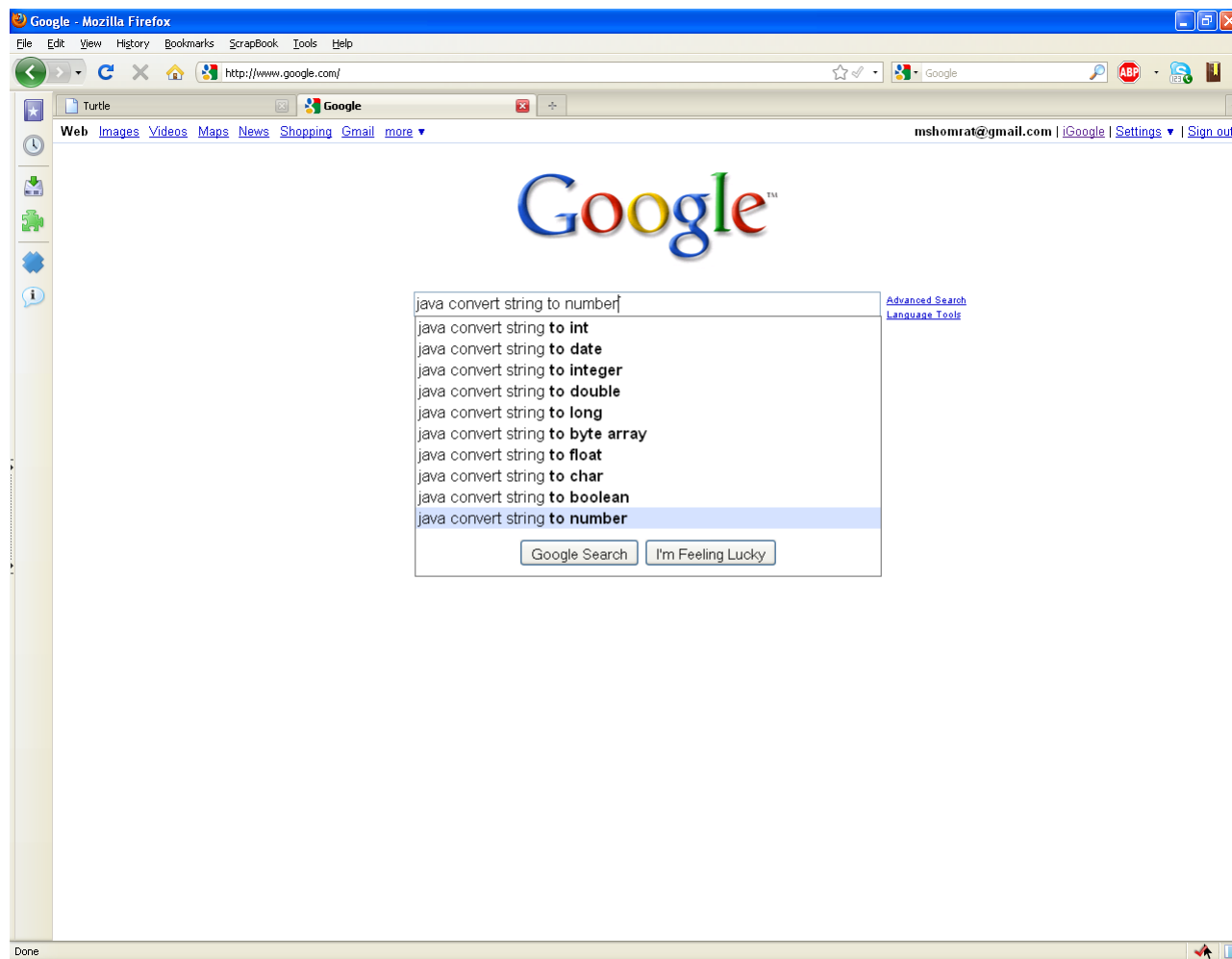
<http://docs.oracle.com/javase/8/docs/api/index.html?java/lang/String.html>

המרת מחרוזות למספרים

- `Long.parseLong`
- `Integer.parseInt`
- `Short.parseShort`
- `Byte.parseByte`
- `Double.parseDouble`
- `Float.parseFloat`
- `Boolean.parseBoolean`

```
public static void main(String[] args) {  
    int i = Integer.parseInt("1"); // i==1  
    double d = Double.parseDouble("-12.45e2"); // d==-1245.0  
}
```

איך נדע אילו פקודות קיימות...?



התשובה



Java Tips


Home ▶ Java SE Tips






Main Menu

- [Home](#)
- [Java Tutorials](#)
- [Book Reviews](#)
- [Java SE Tips](#)
- [Java ME Tips](#)
- [Java EE Tips](#)
- [Other API Tips](#)
- [Java Applications](#)
- [Java Libraries](#)
- [Java Games](#)
- [Sitemap](#)

Java Network

How to convert a string to a number

User Rating:  / 118

Poor      Best

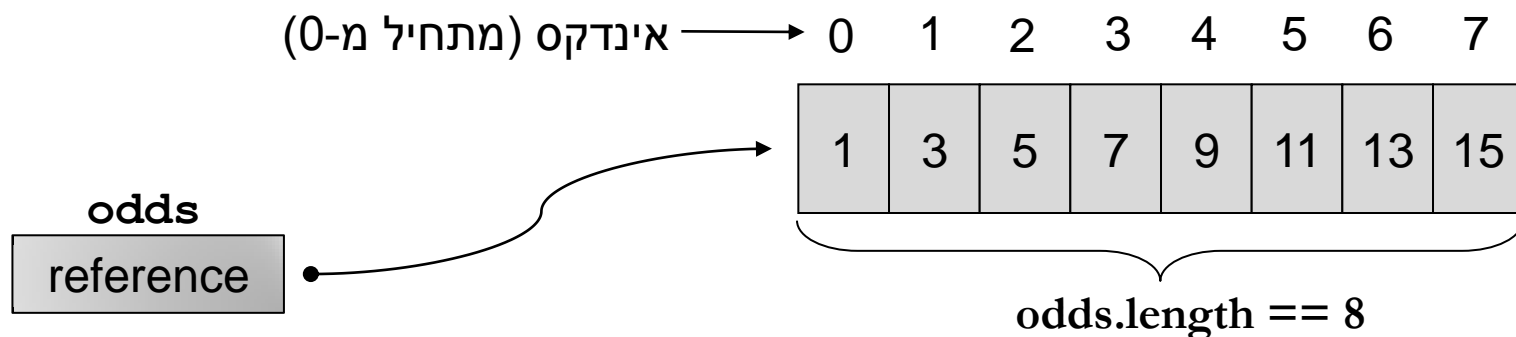
This Java tip illustrates a method of converting a string to a number. This tip is very useful for developer who are expecting a number as a input but recieve the number in a string format. Developers may use this code to convert the retrieved string into number format.

```
byte b = Byte.parseByte("123");
short s = Short.parseShort("123");
int i = Integer.parseInt("123");
long l = Long.parseLong("123");
float f = Float.parseFloat("123.4");
double d = Double.parseDouble("123.4e10");
```

מערכים בקצרה

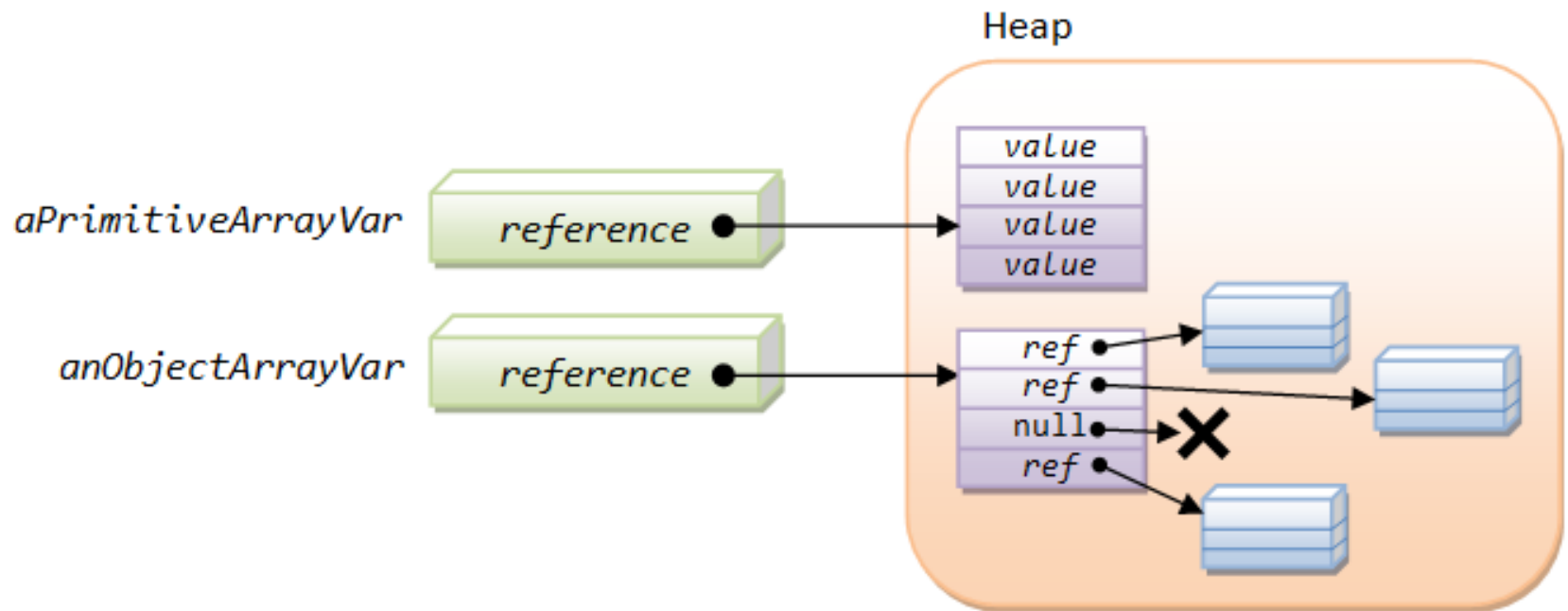
- מבנה נתונים פשוט
- אוסף של פריטים מאותו טיפוס
- גישה באמצעות אינדקס
- נשתמש ב [] לציין טיפוס מסוג מערך.
- הגדרת מערך של int בשם odds:

```
int[] odds = new int[8];
```



- הרחבה על מערכים בתרגול הבא

מערכים

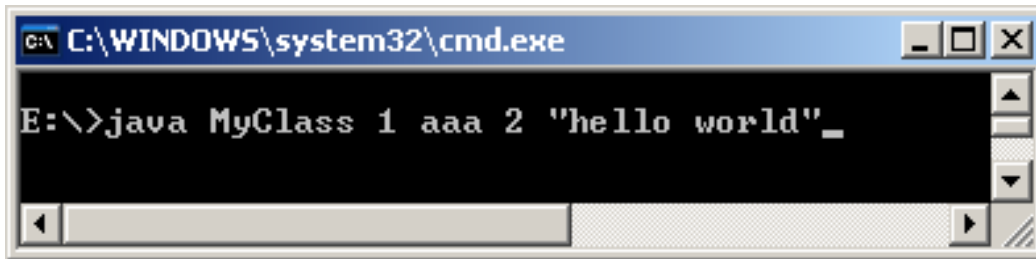


העברת ארגומנטי שורת פקודה לתוכנית

```
public class MyClass {
    public static void main(String[] args) {
        System.out.print(args.length + "\t");
        System.out.print(args[0] + "\t");
        System.out.print(args[args.length - 1]);
    }
}
```

מערך המכיל את
הארגומנטים שהועברו
לתוכנית עם הרצתה

- כיצד מעבירים ארגומנטים לתוכנית? 2 דרכים אפשריות:
- command line



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The command entered is "E:\>java MyClass 1 aaa 2 "hello world"_" and the output is "4 1 hello world".

- Eclipse → Arguments → Run Configurations → Run

Output: 4 1 hello world

מחרוזות ותווים

- כתוב תוכנית שמקבלת תו כארגומנט ומדפיסה:
 - את התו
 - את התו העוקב לו

```
public static void main(String[] args) {  
    char c = args[0].charAt(0);  
    char c1 = (char) (c + 1);  
    System.out.println(c + "\t" + c1);  
}
```

Escape Sequences

תווים מיוחדים

Escape Sequence	Description
<code>\t</code>	Tab
<code>\n</code>	Newline
<code>\r</code>	Carriage return
<code>\'</code>	Single quote
<code>\"</code>	Double quote
<code>\\</code>	Backslash

מחרוזות ותווים

- כתוב תוכנית המקבלת תו מ- {a,b,...,z} ומדפיסה את ה-Uppercase שלו
- נחשב את המיקום של התו ב abc ונמיר אותו לאותו תו (אותו מיקום) ב ABC

```
public static void main(String[] args) {  
    char c = args[0].charAt(0);  
    System.out.println((char) (c - 'a' + 'A'));  
}
```

פתרון א':

דרכים נוספות?

Google ▲

Web [+ Show options...](#)

Results 1 - 10 of

[Converting a String to Upper or Lower Case \(Java Developers ...](#)

Converting a String to **Upper** or **Lower** Case. // **Convert** to **upper case** String **upper** = string.toUpperCase(); // **Convert** to **lower case** String **lower** = string. ...

www.exampledepot.com/egs/java.lang/UpperLower.html - [Cached](#) - [Similar](#) -   

[Converting lowercase to uppercase - Java](#)

6 posts - 3 authors - Last post: 10 Feb 2008

Re: **Converting lowercase to uppercase**. Feb 10th, 2008. Check out the **java** API:

<http://java.sun.com/j2se/1.4.2/docs/...Character.html> ...

www.daniweb.com/forums/thread108555.html - [Cached](#) - [Similar](#) -   

מחרוזות ותווים

• בעזרת String.toUpperCase()

```
public static void main(String[] args) {  
    System.out.println (args[0].toUpperCase ());  
}
```

פתרון ב':

```
public static void main(String[] args) {  
    char c = args[0].charAt(0);  
    System.out.println (Character.toUpperCase (c));  
}
```

פתרון ג':

המרת טיפוסים פרימיטיביים

```

public static void main(String[] args) {
    long l = 20000000000+20000000000; // l == -294967296
    int i = (int) 1.999999999; // i == 1
    float f = (float) 1.999999999; // f == 2
    f = 5/2; // f == 2
    f = (float) (5/2); // f == 2
    f = (float) 5/2; // f == 2.5
    f = 5 / (float) 2; // f == 2.5
    short a = 2;
    ✘ short c = a*a; // compilation error: cannot convert from int to short
}

```

עוד על המרות ב-

http://java.sun.com/docs/books/jls/third_edition/html/conversions.html

הסוף...