كلية الهندسة و
الحوسبة بالقنفذة
قسم الحاسبات

CS1701: البرمجة الحاسوبية بلغة Java

الفصل الدراسي: 2026/27 1ST

1

1

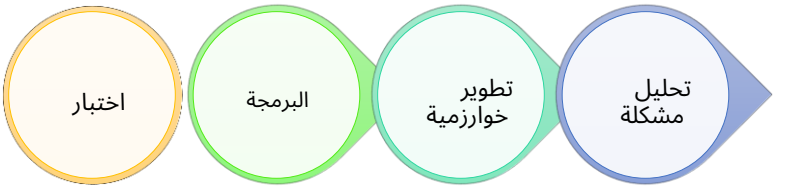
مقدمة إلى الحاسبات ولغة Java

الفصل 1

2

حل المشكلات والخوارزمية

• **حل المشكلات**هو عملية تحديد مشكلة ما، وتطوير خوارزمية، وفي النهاية تنفيذ برنامج حاسوبي



• الـ **خوارزمية** هي مجموعة من الخطوات المحدودة والمتسلسلة وغير الغامضة التي توصف عادة باللغة الطبيعية لحل مشكلة معينة.

3

3

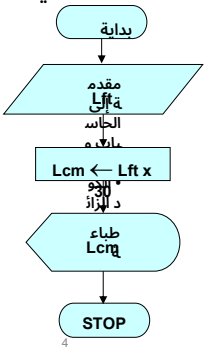
خوارزمية

• أكثر طريقتين شيوعا لتمثيل الخوارزميات هما:

• **المخططات الانسيابية (Flowcharts)**هي تمثيل رسومي للخوارزمية

• **الشيفرة الوهمية (Pseudocode)**هي وصف للتعليمات التي يمكن للإنسان فهمها ومن ثم يمكن للحاسوب تنفيذها لاحقا.

مخطط انسيابي



• **مثال:**

اكتب خوارزمية وارسم مخططا انسيابيا لتحويل الطول من القدم إلى السنتيمتر.

• **الشيفرة الوهمية**:

1. أدخل الطول بالقدم *(Lft)*
2. احسب الطول بالسنتيمتر *(Lcm)* عن طريق ضرب *LFT* في *30*
3. طباعة الطول بالسنتيمتر *(LCM)*

4

4

2

كلمات هذه الصفحة

finite — محدود

instructions — تعليمات

sequential — متسلسل

execute — تنفيذ

unambiguous — غير غامض

Testing — اختبار

graphical representation — تمثيل رسومي

5

6

كلمات هذه الصفحة

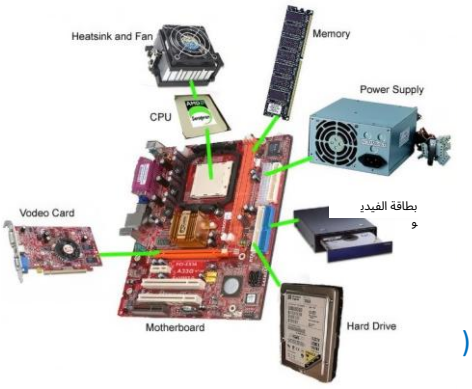
Computer Anatomy — تشريح الحاسوب

Familiarity — الالمام

tangible — ملموس

basics — أساسيات

العتاد



- تمتلك معظم الحواسيب الحديثة مكونات متشابهة بما في ذلك
- **Input devices** (keyboard, mouse, etc.)
- **أجهزة المخرجات** (شاشة العرض، الطابعة، إلخ)
- **وحدة المعالجة المركزية (CPU)**
- **الذاكرة**

7

7

الذاكرة

- تحتفظ الذاكرة بـ
  - البرامج
  - والبيانات التي يعالجها الحاسوب
  - نتائج المعالجة الوسيطة.
- نوعان من الذاكرة
  - الذاكرة الأولية (Primary memory): الذاكرة الافتراضية والأسرع التي يمكن للكمبيوتر الوصول إليها
  - الذاكرة الثانوية/المساعدة

8

8

كلمات هذه الصفحة

modern — حديث

Primary memory — ذاكرة أولية

Central Processing Unit — وحدة المعالجة المركزية

default — افتراضي

holds — تحتفظ بـ

Secondary/auxiliary — ثانوية أو مساعدة

intermediate — وسيطة

RAM ROM

الذاكرة الرئيسية

- يحتوي كل حاسوب على نوعين من الذاكرة الأولية:
  - ذاكرة القراءة فقط (ROM)
  - ذاكرة الوصول العشوائي (RAM)

ROM	RAM
تحتفظ ببرامج وبيانات "غير قابلة للتعديل" تستخدم للعمليات الأساسية مثال: التعليمات الخاصة بكيفية تشغيل تشغيل الحاسوب أو إطفاؤه	تستخدم لتخزين البرنامج الحالي والبيانات التي يستخدمها البرنامج، ونتائج الحسابات الوسيطة
ذاكرة ROM هي ذاكرة غير متطايرة	ذاكرة الوصول العشوائي (RAM) هي ذاكرة متطايرة

9

9

الذاكرة الثانوية

- تستخدم لتخزين كميات ضخمة من البيانات التي تقاس عادة بالجيجابايت

- مثال:
  - محركات الأقراص، والأقراص المدمجة (CDs)، وأقراص الفيديو الرقمية (DVDs)، ومحركات أقراص الفلاش، وما إلى ذلك.

10

10

5

كلمات هذه الصفحة

Read-Only Memory — ذاكرة القراءة فقط

Random Access Memory — ذاكرة الوصول العشوائي

non-volatile — غير متطايرة

volatile — متطايرة

mass data — كميات ضخمة من البيانات

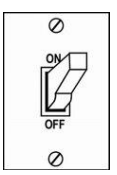
وحدة المعالجة المركزية (CPU)

- يسمى أيضا **المعالج** أو **الشريحة** (على سبيل المثال، معالج Pentium).
- يقوم المعالج **بمعالجة** تعليمات البرنامج.
  - إنه جهاز داخل حاسوبك يقوم بتنفيذ تعليمات البرنامج!

11

11

البتات والبايتات



- الـ **بت** هو رقم ذو قيمة إما **0 أو 1**) **أرقام ثنائية**(
  - وهو أصغر وحدة في ذاكرة الحاسوب
  - من السهل صنع آلات ذات حالتين مستقرتين فقط!
- الـ **بايت** يتكون من **8 بتات**
  - 1 يمكن للبايت تخزين حرف واحد مشفر كسلسلة من الأصفار والآحاد
  - لحسن الحظ، يتم تحويل الأرقام، والحروف، وسلاسل المحارف، والصوت، والفيديو، والبرامج تلقائيا بشكل آلي!

12

12

كلمات هذه الصفحة

chip — شريحة

series — سلسلة

stable states — حالات مستقرة

conversion — تحويل

encoded — مشفر

عناوين ذاكرة RAM

- تنقسم ذاكرة RAM في الحاسوب إلى بايتات بدلا من تقسيمها إلى بتات!

- كل **بايت** في الذاكرة الرئيسية

الذاكرة **تقع في موقع مرقم يسمى** عنوانها *address*.

13

13

مراجعة!

- أي حاسوب يحتوي على:
  - أجهزة إدخال/إخراج
  - وحدة معالجة مركزية (CPU) تعالج المعلومات/البيانات
  - الذاكرة: ذاكرة الوصول العشوائي لتخزين بيانات البرنامج قيد التشغيل و ثانوية لتخزين البيانات بشكل دائم
- **البتات (bits)** هي وحدات ذاكرة لتخزين 1 أو 0
- **البايت (byte)** هو وحدة تخزين مكونة من 8 بتات يمكنها احتواء حرف من لوحة المفاتيح
- كل بايت في الذاكرة الرئيسية له **عنوان**
- لتخزين بيانات أكبر، تستخدم البايتات المتجاورة كوحدة واحدة وحدة ذاكرة
  - يستخدم عنوان أول بايت كعنوان لـ موقع الذاكرة بالكامل.

14

14

كلمات هذه الصفحة

Memory Addresses — عناوين الذاكرة

divided — تنقسم

resides — تقع

Recap — مراجعة

permanently — بشكل دائم

adjacent — متجاورة

البرامج ولغات البرمجة

15

15

البرامج

- الـ **برنامج** هو مجموعة من التعليمات التي يتبعها الحاسوب لـ ينفذها.
- نحن نستخدم البرامج بشكل يومي تقريبا (البريد الإلكتروني، معالجات النصوص، ألعاب الفيديو، أجهزة الصراف الآلي، إلخ).
- يطلق على اتباع التعليمات اسم **التشغيل** أو **تنفيذ** البرنامج.

16

16

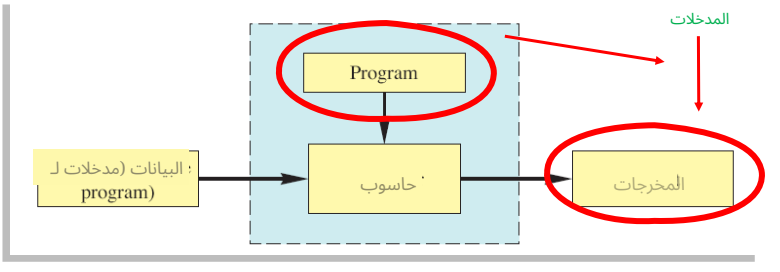
تشغيل برنامج

- عادة، **يتلقى الحاسوب** نوعين من **المدخلات**:
  - البرنامج
  - البيانات التي يحتاجها البرنامج.
- الـ **مخرجات** هي النتيجة (أو النتائج) التي يتم إنتاجها باتباع التعليمات الموجودة في البرنامج.

17

17

تشغيل برنامج - وجهة نظر الحاسوب



ملاحظة: هذا ما يحدث فعليا عند تشغيل برنامج.

18

18

كلمات هذه الصفحة

Normally — عادة

Viewpoint — وجهة نظر

لغات البرمجة

- **لغات البرمجة** هي مجموعة من التعليمات المكتوبة بطريقة يمكن للحاسوب فهمها.

- لماذا نحتاج إلى لغات البرمجة؟
  - جميع مكونات الحاسوب: الـ CPU، والذاكرة، ووحدات الإدخال والإخراج (I/O) تفهم شيفرة آلة تسمى الثنائي (binary).
  - الثنائي هو سلسلة طويلة من الأصفار والآحاد التي لا يمكن للبشر فهمها!
  - إذا، كيف يمكننا جعل الحاسوب ينفذ مهمة معينة؟
  - تساعدنا لغات البرمجة في القيام بذلك من خلال توفير حل وسيط بين الإنسان والآلة.

- **لغات البرمجة** يمكن تصنيفها إلى:
  - اللغات منخفضة المستوى (لغة الآلة) مثل لغة التجميع (assembly language).
  - اللغات عالية المستوى.

19

19

لغات البرمجة عالية المستوى

- اللغات عالية المستوى تعد سهلة نسبيا للقراءة والكتابة والفهم من قبل البشر.
  - على سبيل المثال: Java، وC#، وC++، وVisual Basic، وPython، وRuby.

- لسوء الحظ، لا تفهم أجهزة الحاسوب لغات البرمجة عالية المستوى!

- لذلك، يجب ترجمة برنامج لغة عالية المستوى إلى لغة منخفضة المستوى بواسطة برنامج يسمى مترجم.

20

20

كلمات هذه الصفحة

intermediate solution — حل وسيط

categorized — تصنيف

Low-level — منخفضة المستوى

High-level — عالية المستوى

translated — ترجم

المترجمات

• يقوم الـ **مترجم** بترجمة برنامج من لغة عالية المستوى إلى لغة منخفضة المستوى يمكن للحاسوب تشغيلها.

• أنت تترجم برنامجا عن طريق تشغيل المترجم على نسخة البرنامج المكتوبة بلغة عالية المستوى والتي تسمى **البرنامج المصدري أو الشيفرة المصدرية**.

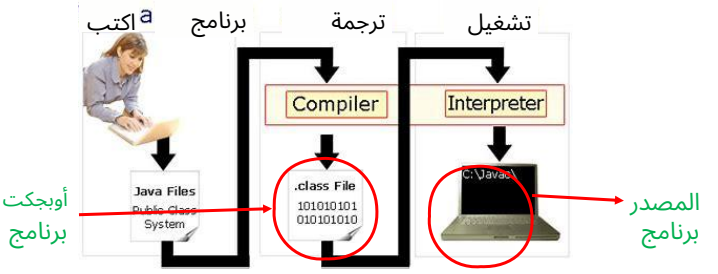
• تنتج المترجمات برامج بلغة الآلة أو لغة التجميع تسمى **برامج كائنية أو شيفرة كائنية**.

•تحدث عملية الترجمة (compilation) مرة واحدة فقط، ويمكن تشغيل برنامج الكائن الناتج بالقدر المطلوب دون الحاجة إلى إعادة ترجمة البرنامج المصدري.

21

21

المترجمات



22

22

11

كلمات هذه الصفحة

source program — برنامج مصدري

object programs — برامج كائنية

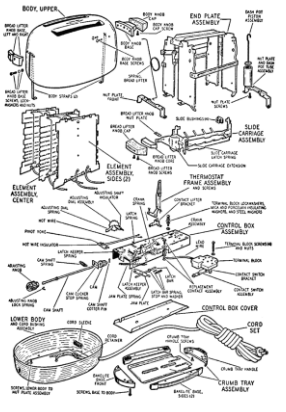
Compilation process — عملية الترجمة

لغة البرمجة Java

23

23

تاريخ Java



- في عام 1991، بدأ **James Gosling and Sun Microsystems** في تصميم لغة للأجهزة المنزلية، مثل محمصة الخبز، وأجهزة التلفاز، وما إلى ذلك.
- تحد كبير، لأن الأجهزة المنزلية تدار بواسطة العديد من الرقاقات (المعالجات) المختلفة!
- كانت البرامج تترجم أولا إلى لغة وسيطة مشتركة بين جميع معالجات الأجهزة.

24

24

12

كلمات هذه الصفحة

home appliances — أجهزة منزلية

Challenging — تحد كبير

تاريخ Java

- ثم كانت ترجم اللغة الوسيطة (المسماة **البايت كود**) إلى لغة الآلة الخاصة بمعالج جهاز معين.

- لم يبد المصنعون معجبين بذلك!

- في عام 1994، أدرك Gosling أن لغته ستكون مثالية لمتصفح ويب يمكنه تشغيل برامج عبر الإنترنت.
  - أنتجت شركة Sun المتصفح المعروف اليوم باسم HotJava.

25

25

تطبيقات Java والـ Applets

- نوعان من برامج Java:
  - التطبيقات (Applications)
    - برامج اعتيادية
    - مخصصة للتشغيل على حاسوبك
  - تطبيقات Applets
    - تطبيقات صغيرة
    - مصممة ليتم إرسالها إلى موقع آخر على الإنترنت وتشغيلها هناك
    - غير مدعوم من قبل العديد من متصفحات الويب اليوم لصالح HTML5 و JavaScript

26

26

كلمات هذه الصفحة

bytecode — بايت كود

Applications — تطبيقات

manufacturers — المصنعون

Applets — تطبيقات صغيرة

impressed — معجب

supported — مدعوم

ideal — مثالية

بيئة تطوير Java

- هناك أربع مراحل لإنشاء وتنفيذ تطبيق Java:

1. التحرير،
2. الترجمة،
3. التحميل،
4. التفسير والتنفيذ

27

27

الخطوة 1: إنشاء البرامج وتحريرها

- تتضمن هذه الخطوة استخدام **محرر** لكتابة برنامج Java (الشيفرة المصدرية)، ثم حفظه على جهاز تخزين ثانوي، مثل القرص الصلب.
  - محررات معروفة: Notepad على Windows و TextEdit على OS X

- **توفر بيئات التطوير المتكاملة (IDEs)** أدوات تدعم عملية تطوير البرمجيات، مثل المحررات، وأدوات تنقيح الأخطاء (debuggers) لتحديد الأخطاء المنطقية والمزيد.
  - بيئات Java IDEs الشائعة:
    - Eclipse (www.eclipse.org)
    - NetBeans (www.netbeans.org)
    - IntelliJ IDEA (www.jetbrains.com)
    - VSCode (code.visualstudio.com -- ليس فقط لـ Java!)

28

28

كلمات هذه الصفحة

Development Environment — بيئة تطوير

phases — مراحل

editor — محرر

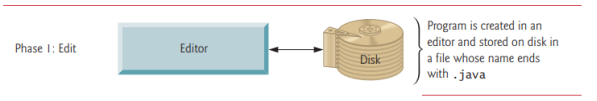
Integrated development environments — بيئات التطوير المتكاملة

debuggers — أدوات تنقيح الأخطاء

logic errors — أخطاء منطقية

الخطوة 1: إنشاء البرامج وتحريرها

• تنتهي جميع ملفات شيفرة Java المصدرية بالامتداد **.java**.



الشكل 1.6 بيئة تطوير Java النموذجية - مرحلة التحرير.

29

29

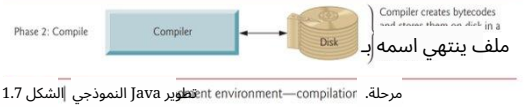
الخطوة 2: ترجمة برنامج Java إلى شيفرات

• لا **يقوم مترجم** Java **بترجمة** برنامج Java إلى لغة الآلة الخاصة بحاسوب معين.

• بدلا من ذلك، **فإنه يترجم برنامج Java إلى البايت كود**.

• **البايت كود** هو لغة الآلة الخاصة بحاسوب افتراضي (أو **مفسر**) يسمى **آلة Java الافتراضية (JVM)**.

• يحتوي ملف**البايت كود** على .class كلاحقة للملف



مرحلة. ent environment—compilatior

الشكل 1.7 تطوير Java النموذجي

30

30

كلمات هذه الصفحة

extension — امتداد

hypothetical — افتراضي

الخطوة رقم 2: ترجمة برنامج Java إلى البايت كود (byte-code)

- يمكن استخدام البايت كود على أي حاسوب أو منصة تحتوي على *JVM*.
  - لذلك، تعد Java لغة برمجة مستقلة عن المنصة (قابلة للنقل)

- يمكن إرسال البايت كود عبر الإنترنت واستخدامه في أي مكان في العالم.

- وهذا ما يجعل Java مناسبة لتطبيقات الإنترنت.

31

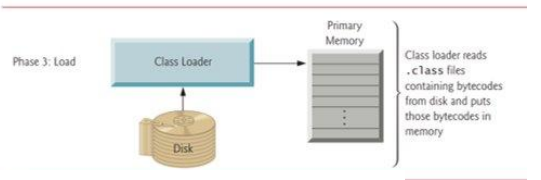
31

الخطوة رقم 3: تحميل البرنامج إلى الذاكرة

- **تقوم الـ JVM** بوضع البرنامج في الذاكرة لتنفيذه—تعرف هذه العملية باسم التحميل

- يقوم **محمل الكلاسات (class loader) الخاص بالـ JVM** بأخذ **.class** الملفات التي تحتوي على البايت كود (bytecode) الخاصة بالبرنامج ونقلها إلى الذاكرة الرئيسية.

- **محمل الكلاسات** يربط الكلاسات ببعضها تلقائيا.



الشكل 1.8 مرحلة تحميل Java النموذجية.

32

32

كلمات هذه الصفحة

platform-independent — مستقلة عن المنصة

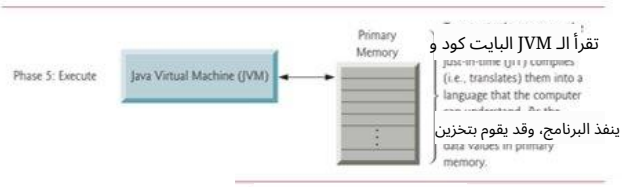
portable — قابلة للنقل

class loader — محمل الكلاسات

الخطوة 4: التفسير والتنفيذ

- **آلة Java الافتراضية (JVM) تترجم** كل **بايت كود** وتنفذ **التعليمات** الناتجة **بلغة الآلة**.

- تنفذ معظم برامج Java اليوم باستخدام **مترجم فوري أو ما يعرف بـ *JIT* مترجم.**



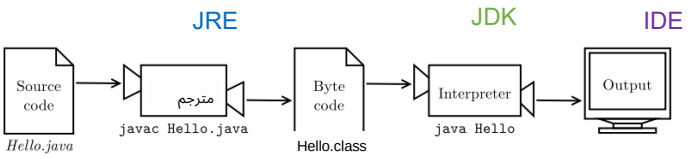
الشكل 1.10 بيئة تطوير Java النموذجيةا

مرحلة تنفيذ البيئة.

33

33

مراجعة: ترجمة وتشغيل برنامج Java (الصورة العامة)



الشكل 1.3: عملية ترجمة وتشغيل برنامج Java.

34

34

17

كلمات هذه الصفحة

Interpreting — تفسير

Overall Picture — الصورة العامة

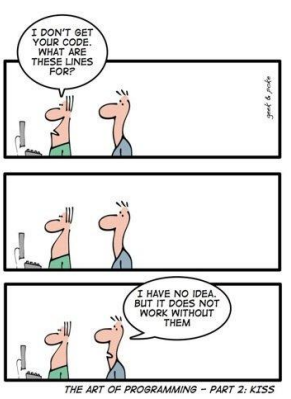
Just-In-Time — مترجم فوري

أساسيات البرمجة

35

35

ما هي البرمجة؟



- يمكننا اعتبار البرمجة عملية تقسيم مهمة كبيرة ومعقدة إلى مهام فرعية أصغر فأصغر.

- تستمر العملية حتى تصبح المهام الفرعية بسيطة بما يكفي لتنفيذها باستخدام الدوائر الإلكترونية التي توفرها الأجهزة.

36

36

كلمات هذه الصفحة

breaking down — تقسيم

subtasks — مهام فرعية

complex — معقد

electronic circuits — دوائر إلكترونية

ما هي البرمجة؟

- بعض التعليمات الأساسية التي يمكن العثور عليها في كل لغة برمجة:

- **الإدخال:** الحصول على البيانات من لوحة المفاتيح، أو ملف، أو مستشعر، أو أي جهاز آخر.
- **الإخراج:** عرض البيانات على الشاشة أو إرسالها إلى ملف أو جهاز آخر.
- **الرياضيات:** تنفيذ العمليات الحسابية الأساسية مثل الجمع والقسمة.
- **اتخاذ القرار:** التحقق من شروط معينة وتنفيذ الشيفرة المناسبة.
- **التكرار:** تنفيذ إجراء ما بشكل متكرر، عادة مع وجود بعض الاختلافات.

37

37

أول تطبيق بلغة Java

القائمة 1.1 A برنامج Java نموذجي

```
import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Hello out there.");
        System.out.println("I will add two numbers for you. ");
        System.out.println("Enter two whole numbers on a line:");

        int n1, n2;

        Keyboard keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("مجموع هذين الرقمين هو" + n1 + n2);
    }
}
```

يجلبُ كلاس Scanner من java.util

اسم الكلاس من اختيارك. "هذا البرنامج should be in a file named FirstProgram.java"

يرسل المخرجات إلى الشاشة

Saves that n1 and n2 are variables

التي تحفظ أعدادا صحيحة

Readies the program for keyboard input

ماسح ضوئي

يقرأ عددا صحيحا واحدا from the keyboard

System.out.println("مجموع هذين الرقمين هو" + n1 + n2);

38

38

19

كلمات هذه الصفحة

mathematical operations — عمليات حسابية

addition — جمع

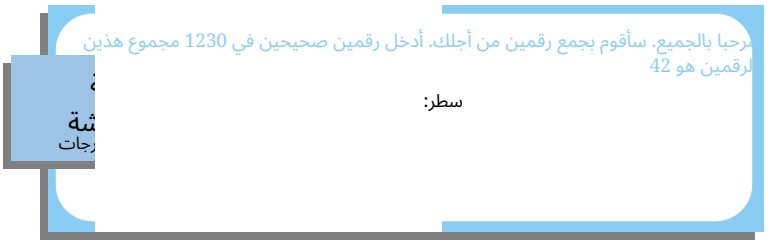
division — قسمة

repetition — تكرار

repeatedly — بشكل متكرر

variation — اختلافات

أول تطبيق بلغة Java



39

39

بعض مصطلحات Java

- يسمى الشخص الذي يكتب البرنامج بـ **مبرمج** !
- الشخص الذي يتفاعل مع البرنامج يسمى الـ **مستخدم**.
- الـ ***package*** هي مكتبة من الكلاسات التي تم تعريفها مسبقا.

```
import java.util.Scanner;
```

40

40

كلمات هذه الصفحة

Terminology — مصطلحات

package — حزمة برمجية

interacts — يتفاعل

defined — معرفة

بعض مصطلحات Java

- تسمى العناصر الموجودة داخل الأقواس **أرجومنت** *(argument)* وهي توفر المعلومات التي تحتاجها الميثودات.

- الـ **متغير** هو شيء يمكنه تخزين البيانات.

- تسمى التعليمات الموجهة للحاسوب **عبارة؛** وتنتهي بفاصلة منقوطة **;**

- تسمى القواعد النحوية للغة البرمجة **بنية (syntax)** اللغة.

41

41

الطباعة على الشاشة

```
System.out.println ("Whatever you want to print");
```

- `System.out` هو أوبجكت (object) لإرسال المخرجات إلى الشاشة.
- `println` هو ميثود (method) لطباعة أي شيء موجود بين قوسين على الشاشة.

- يقوم الأوبجكت بتنفيذ إجراء ما عندما تقوم بـ استدعاء أو طلب أحد الميثودات الخاصة به.

```
objectName.methodName(argumentsTheMethodNeeds);
```

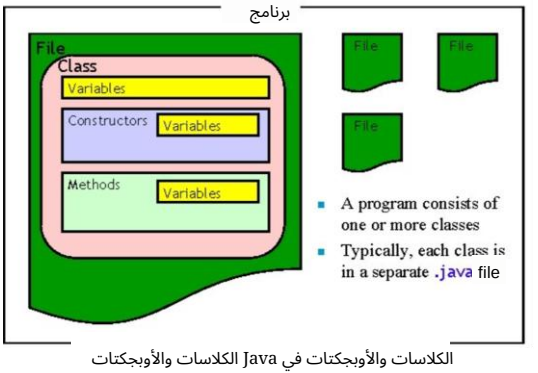
42

42

كلمات هذه الصفحة

parentheses — أقواس	**semicolon** — فاصلة منقوطة
argument — أرجومنت	**grammar rules** — قواعد نحوية
variable — متغير	**syntax** — بنية
statement — عبارة برمجية	**invoke** — استدعاء

ترجمة وتشغيل برنامج أو كلاس Java



الكلاسات والأوبجكتات في Java الكلاسات والأوبجكتات

• يتكون برنامج Java من كلاس واحد أو أكثر، والتي يجب ترجمتها قبل تشغيل البرنامج.

• لا تحتاج إلى ترجمة الكلاسات المرافقة لـ Java (مثل `System` و `Scanner`).

• يجب أن يكون كل كلاس في ملف منفصل.

43

43

ترجمة وتشغيل برنامج أو كلاس Java

• يجب أن يكون اسم الملف مطابقا لـ اسم الكلاس.

• س يحتوي الكلاس المراد تشغيله على العبارة (يجب أن يتضمن كلاس واحد فقط الـ main method)

```
public static void main(String[] args)
```

سيصبح كل هذا أكثر
وضوحا عندما نبدأ في كتابة
كلاسات Java!

44

44

كلمات هذه الصفحة

accompany — مرافقة

الاختبار وتصحيح الأخطاء

- إن أفضل طريقة لكتابة برنامج صحيح هي التصميم الدقيق للأوبجكتات الضرورية والخوارزميات الخاصة بميثودات تلك الأوبجكتات.

- ثم تقوم بترجمة كل شيء بعناية إلى لغة برمجة مثل Java.

- لذا، فإن القضاء على الأخطاء يعني تجنبها في المقام الأول :)

- اختبر برنامجك باستخدام حالات اختبار مناسبة (بعضها تكون فيه الإجابة معروفة)، واكتشف أي أخطاء وقم بإصلاحها، ثم أعد الاختبار.

45

45

الأخطاء

- يسمى الخطأ في البرنامج *bug*.

- تسمى عملية إزالة الأخطاء تصحيح الأخطاء.

- ثلاثة أنواع من الأخطاء
  - أخطاء الصيغة
  - أخطاء وقت التشغيل
  - أخطاء منطقية

46

46

كلمات هذه الصفحة

Debugging — تصحيح الأخطاء

test cases — حالات اختبار

design — تصميم

bug — خطأ برمجيا

eliminate — القضاء على

Runtime Errors — أخطاء وقت التشغيل

أخطاء الصيغة

- إن صيغة لغة البرمجة هي مجموعة القواعد النحوية للطريقة الصحيحة لكتابة البرنامج.

- وبالتالي، فإن أخطاء الصيغة هي الأخطاء النحوية في البرنامج

- يكتشف المترجم (compiler) أخطاء الصيغة ويطبع رسالة خطأ.

- مثال: استخدام نقطة في موضع يتوقع فيه البرنامج فاصلة

47

47

أخطاء وقت التشغيل

- الأخطاء التي يتم اكتشافها أثناء تشغيل برنامجك، ولكن ليس أثناء الترجمة

- عندما يكتشف الحاسوب خطأ ما، فإنه ينهي البرنامج ويطبع رسالة خطأ.

- مثال: محاولة القسمة على 0

48

48

كلمات هذه الصفحة

catches — يكتشف

terminates — ينهي

detected — مكتشف

أخطاء المنطق

- الأخطاء التي لا يتم اكتشافها أثناء الترجمة أو أثناء التشغيل، ولكنها تتسبب في إعطاء البرنامج نتائج غير صحيحة.

- أخطاء المنطق لن تظهر لك أي رسائل خطأ. ولهذا السبب، تعد أخطاء المنطق أصعب أنواع الأخطاء التي يمكن تحديد موقعها.

- مثال: محاولة حساب درجة الحرارة بالفهرنهايت من درجة الحرارة بالسيلزيوس عن طريق الضرب في 9/5 وإضافة 23 بدلا من 32.

49

موارد إضافية

- Harmash.com (باللغة العربية)

https://harmash.com/home/

- رواق (بالعربية)https://www.rwaq.org/

- w3resource (بالإنجليزية)https://www.w3resource.com/

50