- CS117 OOP
- Chapter 2
- Introduction to Java

By Dr. Paween Khoenkaw
Computer Science MJU

Introduction to Java

Outline

-Computer Programming

-Computer Programming model

-Nuts and Bolts of computer programming

-Object Oriented Programming model

-Basic Java programming

-Basic Java Swing framework

What is computer program ?

Introduction to Java

Classic computer program look like this

http://www.youtube.com/watch?v=qux3fLCf2FU

Simple building part but hard to combine them together

Introduction to Java

Modern computer program look like this

http://www.youtube.com/watch?v=8u1qXDM8W94

You learn the building part, put them together is simple

Introduction to Java

History of Java

Java (*this is what you need to know for this course*)

- A complete programming language developed by Sun
- Can be used to develop either web based or stand-alone software
- Many pre-created code libraries available
- For more complex and powerful programs

Developed by Sun from 1991 and first released in 1995

Introduction to Java
History of Java

James Arthur Gosling

[Sun Microsystems]
[Oracle Corporation]
[Google]
[Liquid Robotics]
[Typesafe Inc.]

Computer Program

A computer program for calculating 1 + 2

In machine language

•Store number 1 at memory location A	MOV A,1
•Store number 2 at memory location B	MOV B,2
•Add contents of Location A & B	ADD A,B
•Store results	STORE C
•Operating system call	INT 21

b4 02 b2 49 cd 21
b2 03 cd 21 b2 55
cd 21 b4 4c cd 21

Computer Program

A computer program for calculating 1 + 2

In C

printf(1+2); → Complier → Assembler →

```
MOV A,1
MOV B,2
ADD A,B
STORE C
INT 21
```

```
b4 02 b2 49 cd 21 b2
03 cd 21 b2 55 cd 21
b4 4c cd 21
```

Computer Program

A computer program for calculating 1 + 2

In machine language

- Store number 1 at memory location A — MOV A,1
- Store number 2 at memory location B — MOV B,2
- Add contents of Location A & B — ADD A,B
- Store results — STORE C
- Operating system call — INT 21

CPU

OS

CPU

+

OS

=

Platform

Computer Platforms

Computer Platforms

ADD A,B

PLUS A,B

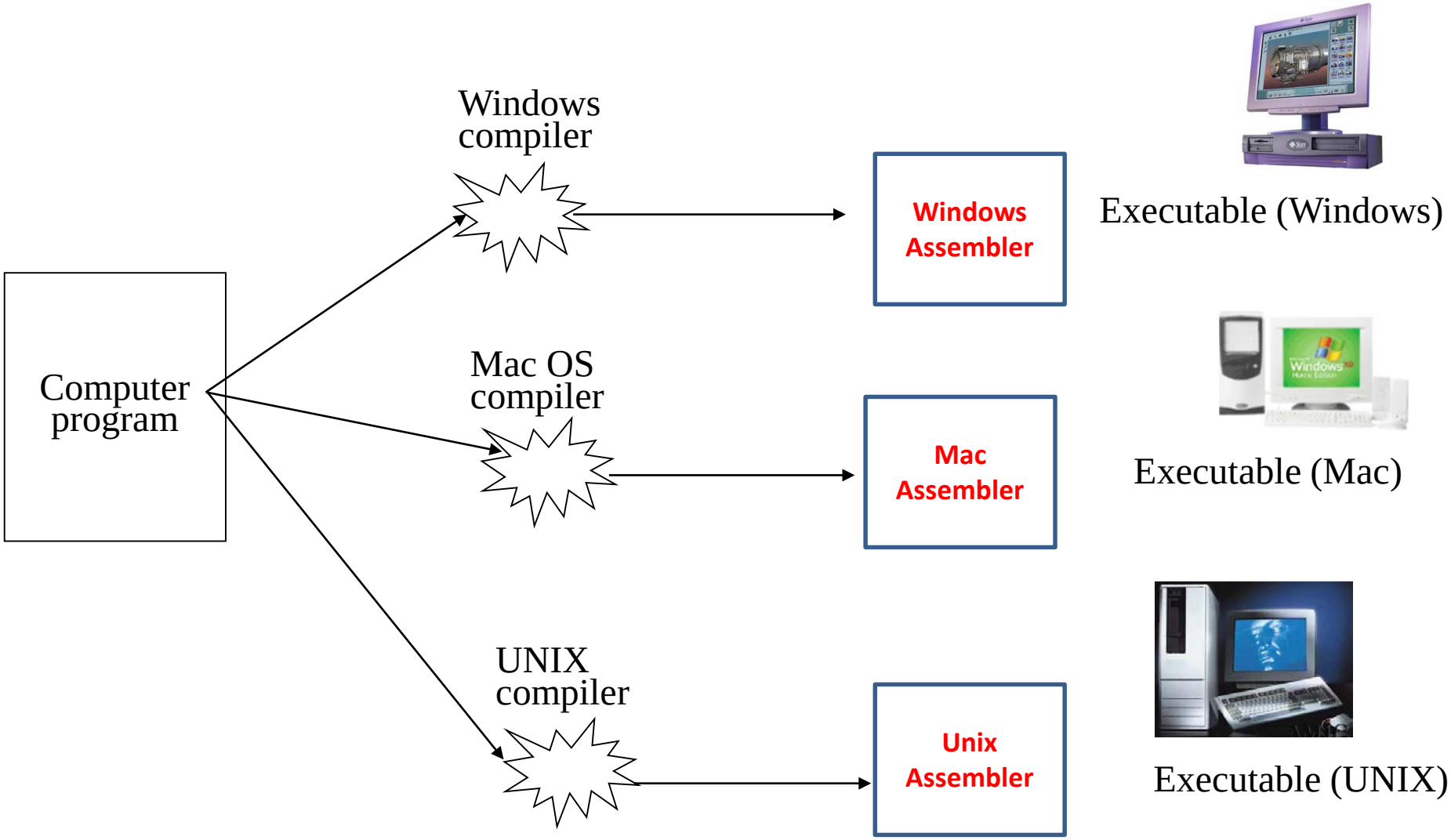
INT 21

CALL 21

RUN 21

Introduction to Java

History of Java



Introduction to Java

History of Java

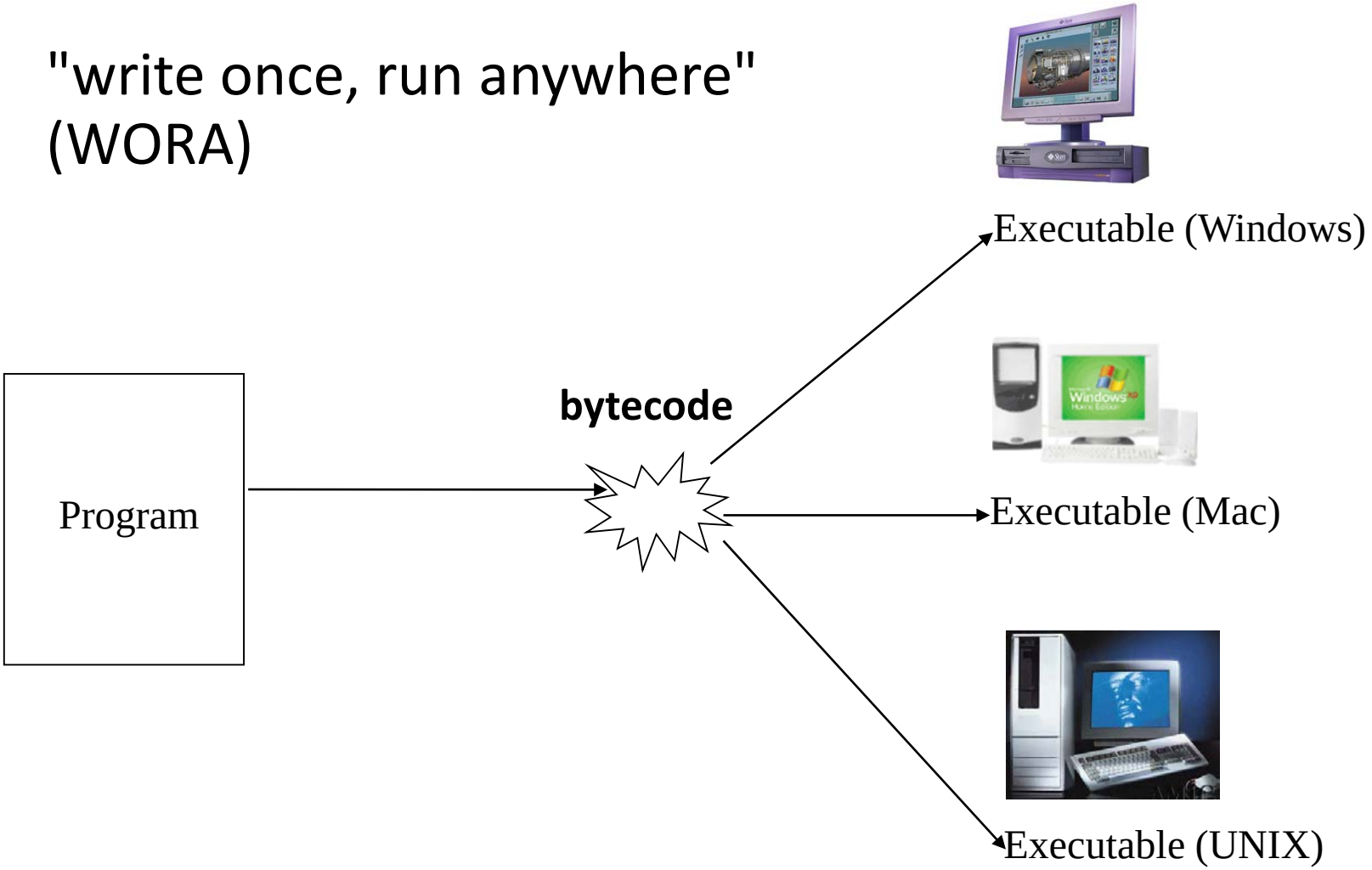
Introduction to Java

History of Java

Introduction to Java

History of Java

"write once, run anywhere" (WORA)



Introduction to Java

History of Java

Sun Microsystems funded an internal research project “Green” to investigate this opportunity.

Result: A programming language called “Oak”

A develop a language for digital devices such as set-top boxes, televisions etc.

Introduction to Java

History of Java

Problem: There was already a programming language called Oak.

The “Green” team met at a local coffee shop to come up with another name...

Introduction to Java
History of Java

Java coffee refers to coffee beans produced in the Indonesian island of Java.

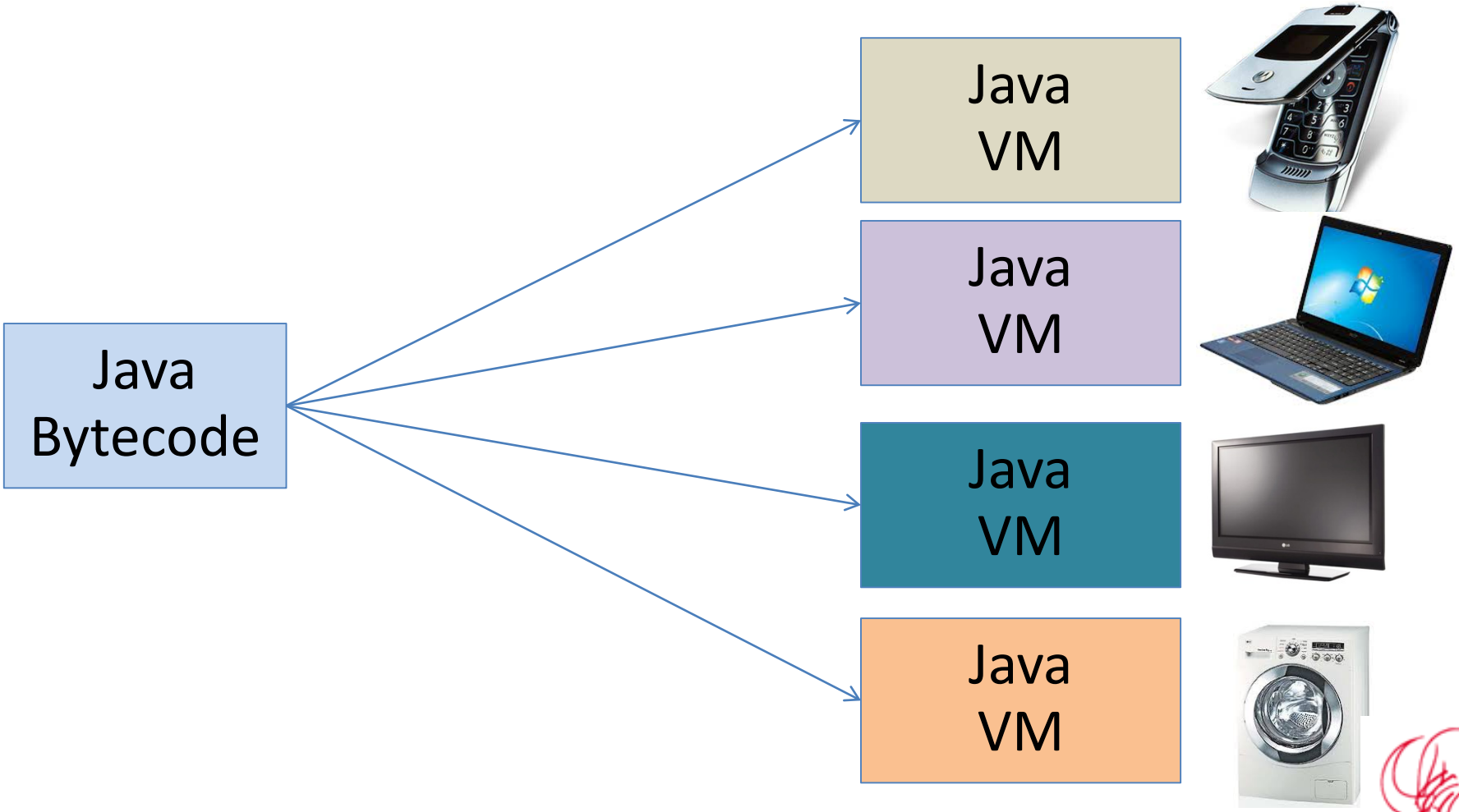
Introduction to Java

How Java works

Stage 1

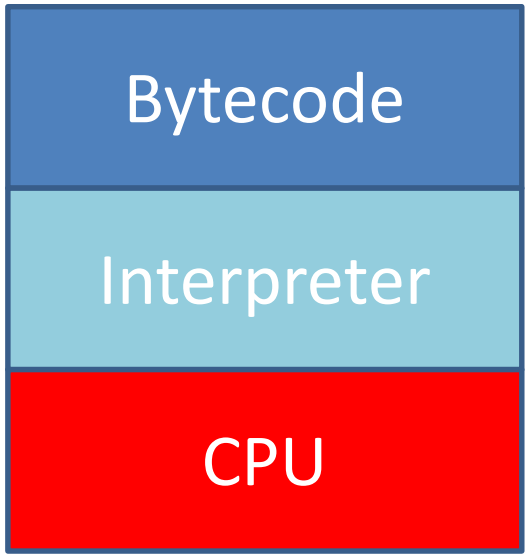
Introduction to Java

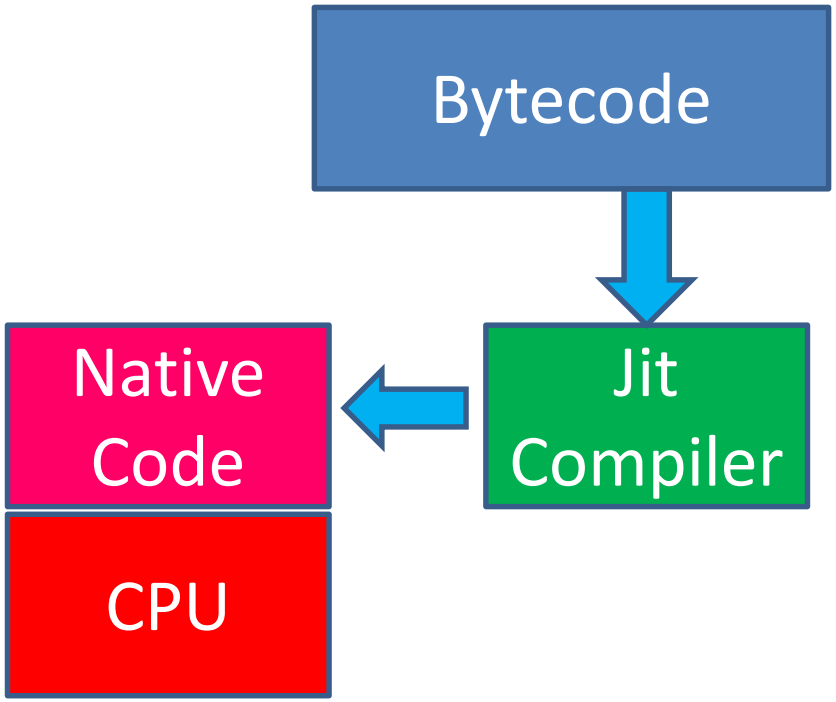
How Java works



Stage 2

Introduction to Java: JVM





Interpreter

1995

Simple, less memory

10-100x slower than ‘C’

Just-In-Time Compiler

1998

Complex, more memory

Almost same as ‘C’

Introduction to Java

Java 6+ JDK (Java Development Kit), Standard Edition includes:

J*D*K (Java development kit) – for *developing* Java software (creating Java programs.

J*R*E (Java Runtime environment) – only good for *running* pre-created Java programs.

Java Plug-in – a special version of the JRE designed to run through web browsers.

Introduction to Java

Java Standard Edition (JAVA SE)

Java Enterprise Edition (JAVA EE)

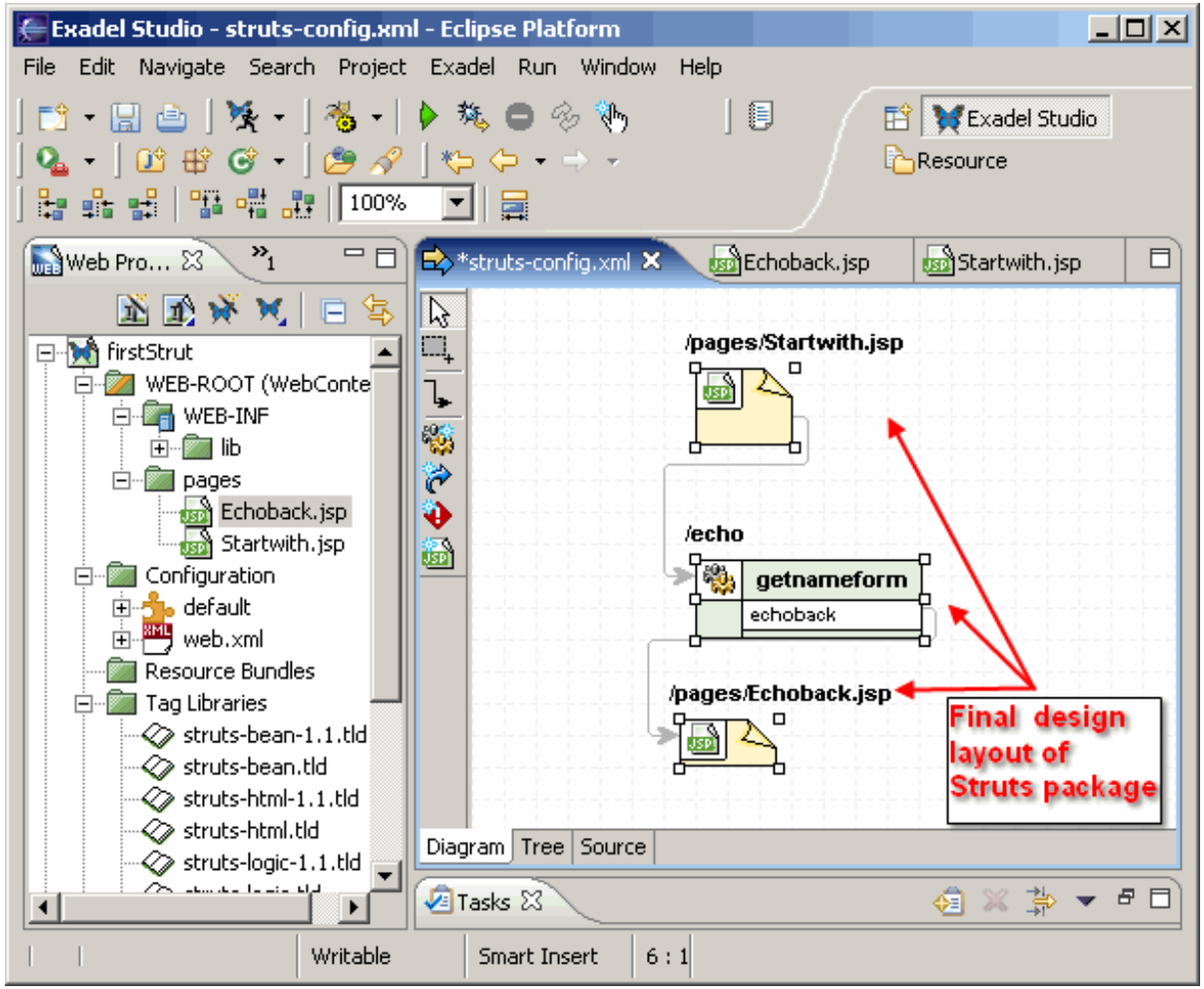
Java Micro Edition (JAVA ME)

Introduction to Java: History

1. JDK Alpha and Beta (1995)
2. JDK 1.0 (23rd Jan, 1996)
3. JDK 1.1 (19th Feb, 1997)
4. J2SE 1.2 (8th Dec, 1998)
5. J2SE 1.3 (8th May, 2000)
6. J2SE 1.4 (6th Feb, 2002)
7. J2SE 5.0 (30th Sep, 2004)
8. Java SE 6 (11th Dec, 2006)
9. Java SE 7 (28th July, 2011)
10. Java SE 8 (18th March, 2014)
11. Java SE 9 (21th September, 2017)
12. Java SE 10 (...)

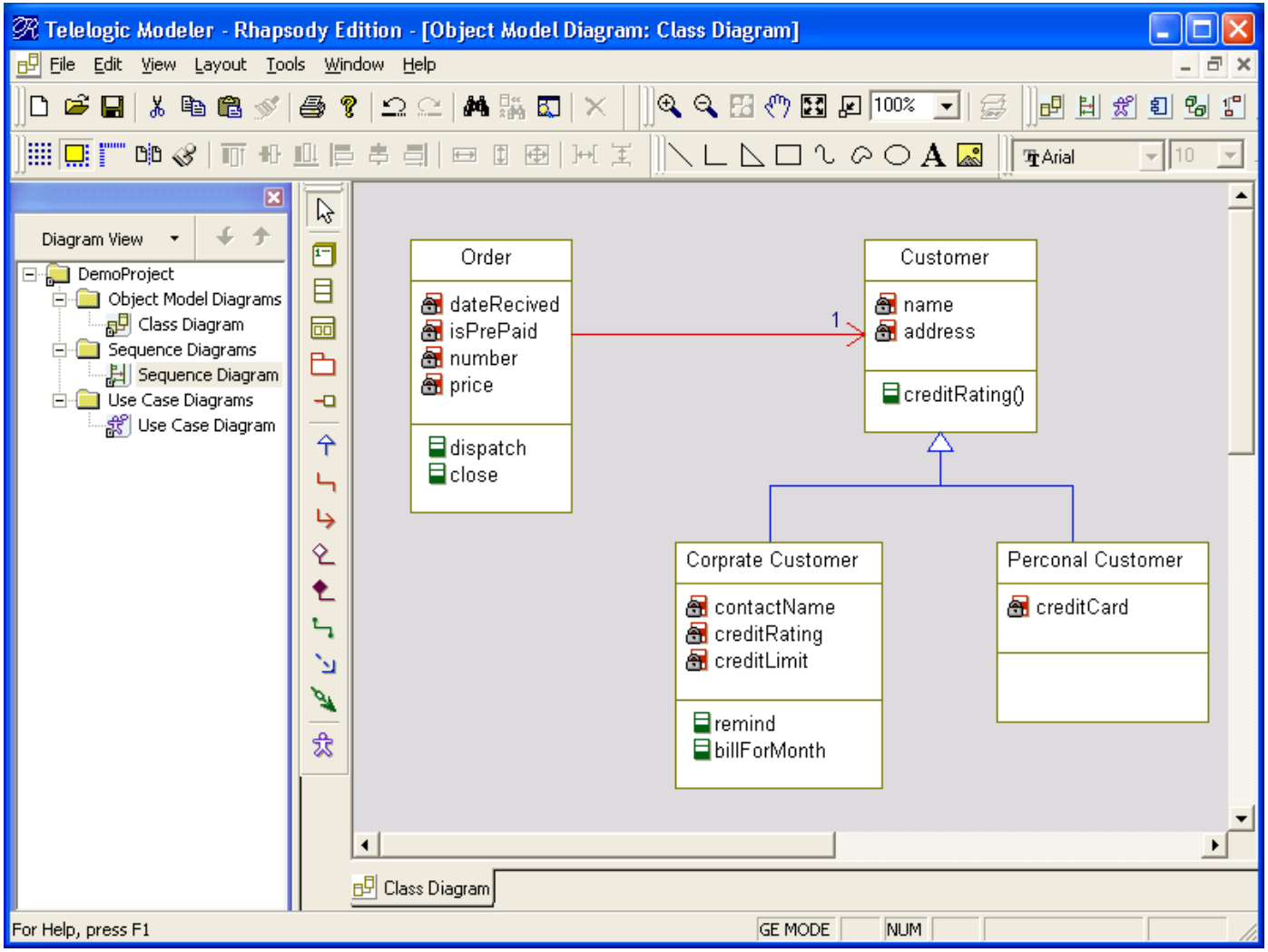
Introduction to Java

Eclipse[1]: started as a programming environment created by IBM for developing Java programs. The program Eclipse was itself written in Java.



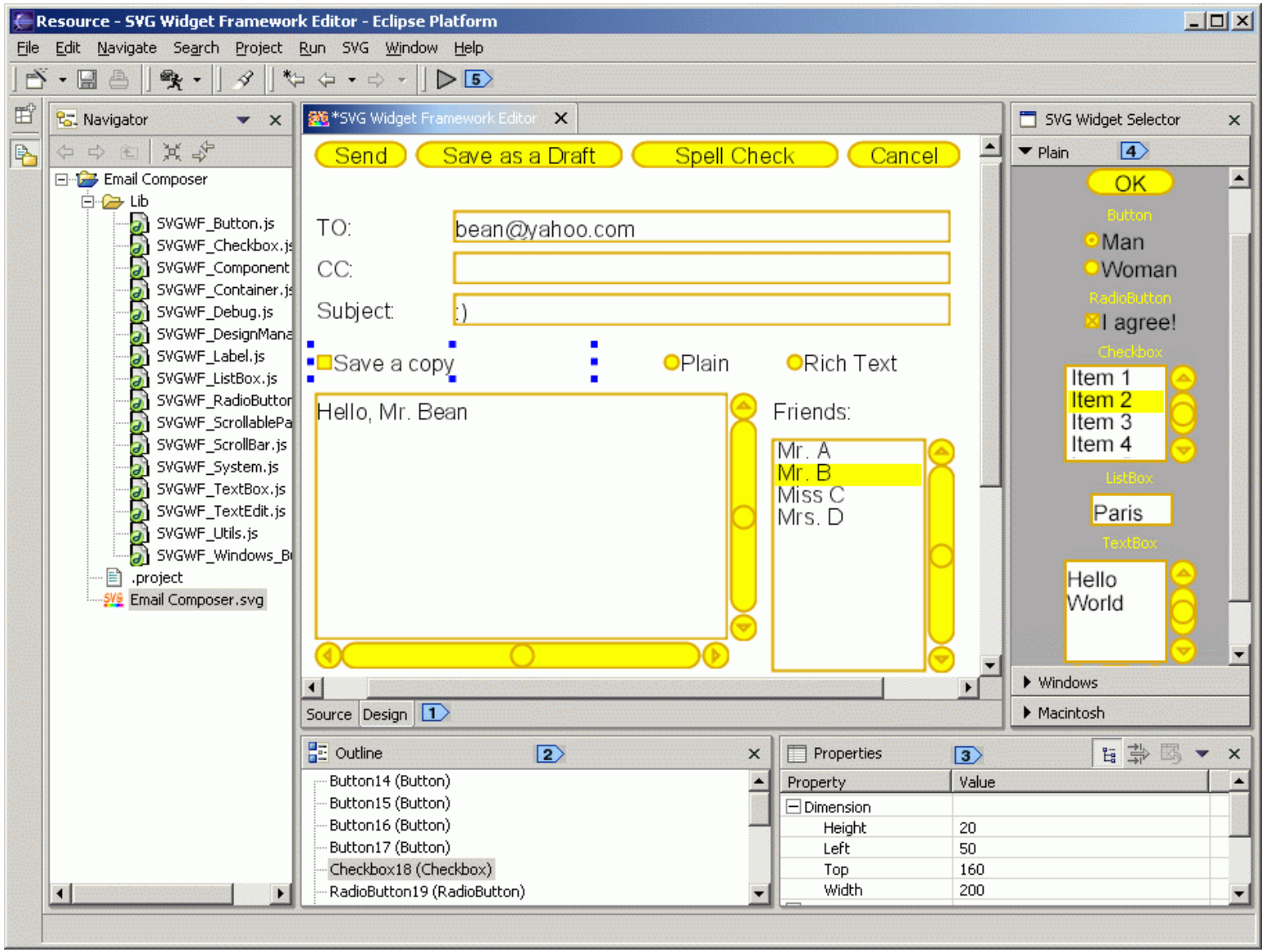
Introduction to Java

Eclipse: Visual Paradigm



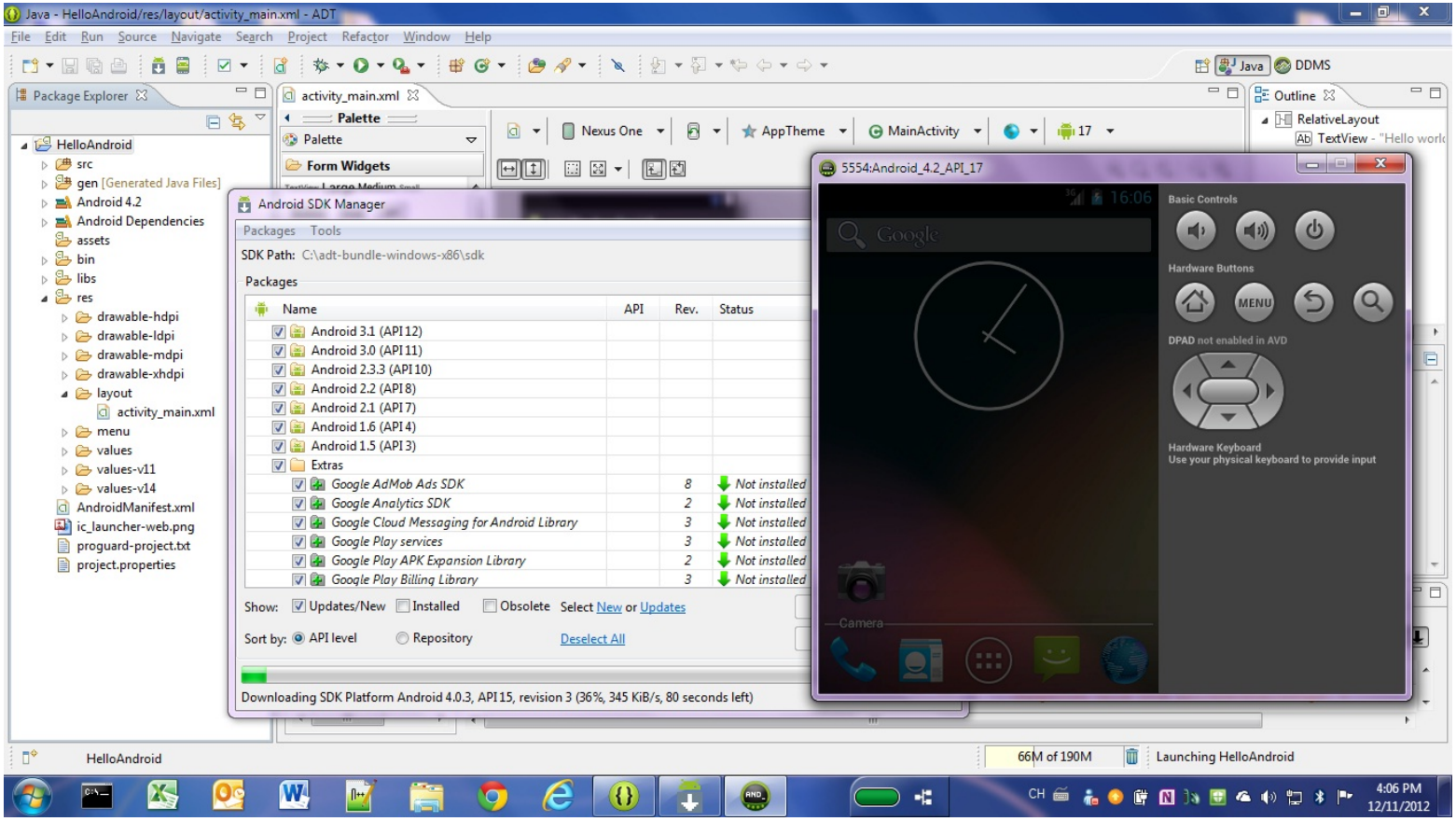
Introduction to Java

Eclipse: Web



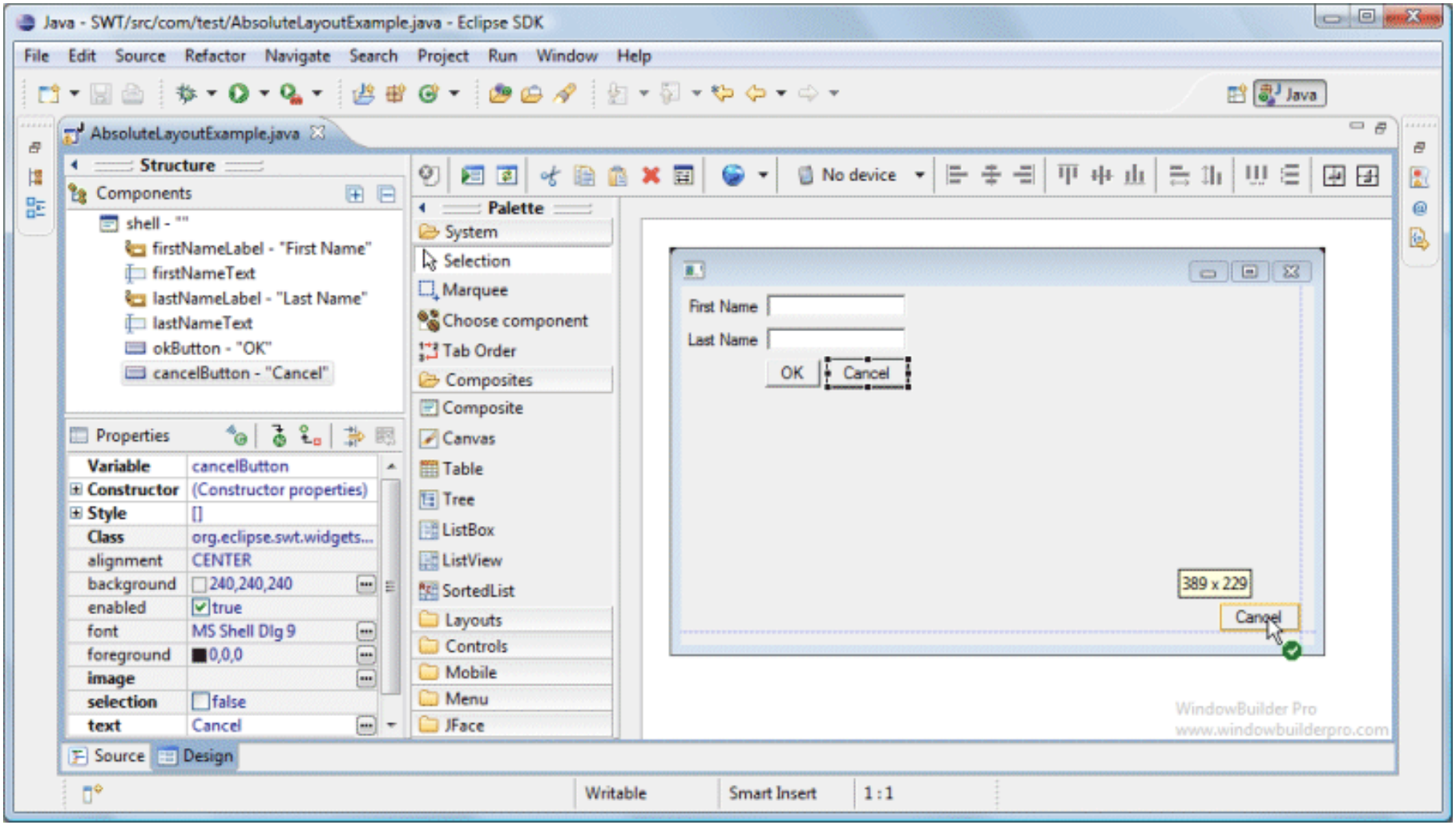
Introduction to Java

Eclipse: Android



Introduction to Java

Eclipse: WindowsBuilder



Introduction to Java:Hello world

```
public class hello_java {

public static void main(String[] args) {

System.out.println("Hello world!");

}

}
```

Introduction to Java: Variable

The variable

Introduction to Java: Variable

Type	Description
byte	8 bit signed integer
short	16 bit signed integer
int	32 bit signed integer
long	64 bit signed integer
float	32 bit signed real number
double	64 bit signed real number
char	16 bit Unicode character (ASCII and beyond)
boolean	1 bit true or false value
String	A sequence of characters between double quotes ("")

Introduction to Java: Variable

PRIMITIVE DATA TYPE in JAVA

Type	Contains	Default	Size	Range
byte	Signed integer	0	8 bits	-128 to 127
short	Signed integer	0	16 bits	-32768 to 32767
int	Signed integer	0	32 bits	-2147483648 to 2147483647
float	IEEE 754 floating point	0.0f	32 bits	±1.4E-45 to ±3.4028235E+38
long	Signed integer	0L	64 bits	-9223372036854775808 to 9223372036854775807
double	IEEE 754 floating point	0.0d	64 bits	±4.9E-324 to ±1.7976931348623157E+308
boolean	true or false	FALSE	1 bit	NA
char	Unicode character	'\u0000'	16 bits	\u0000 to \uFFFF

Introduction to Java: Variable

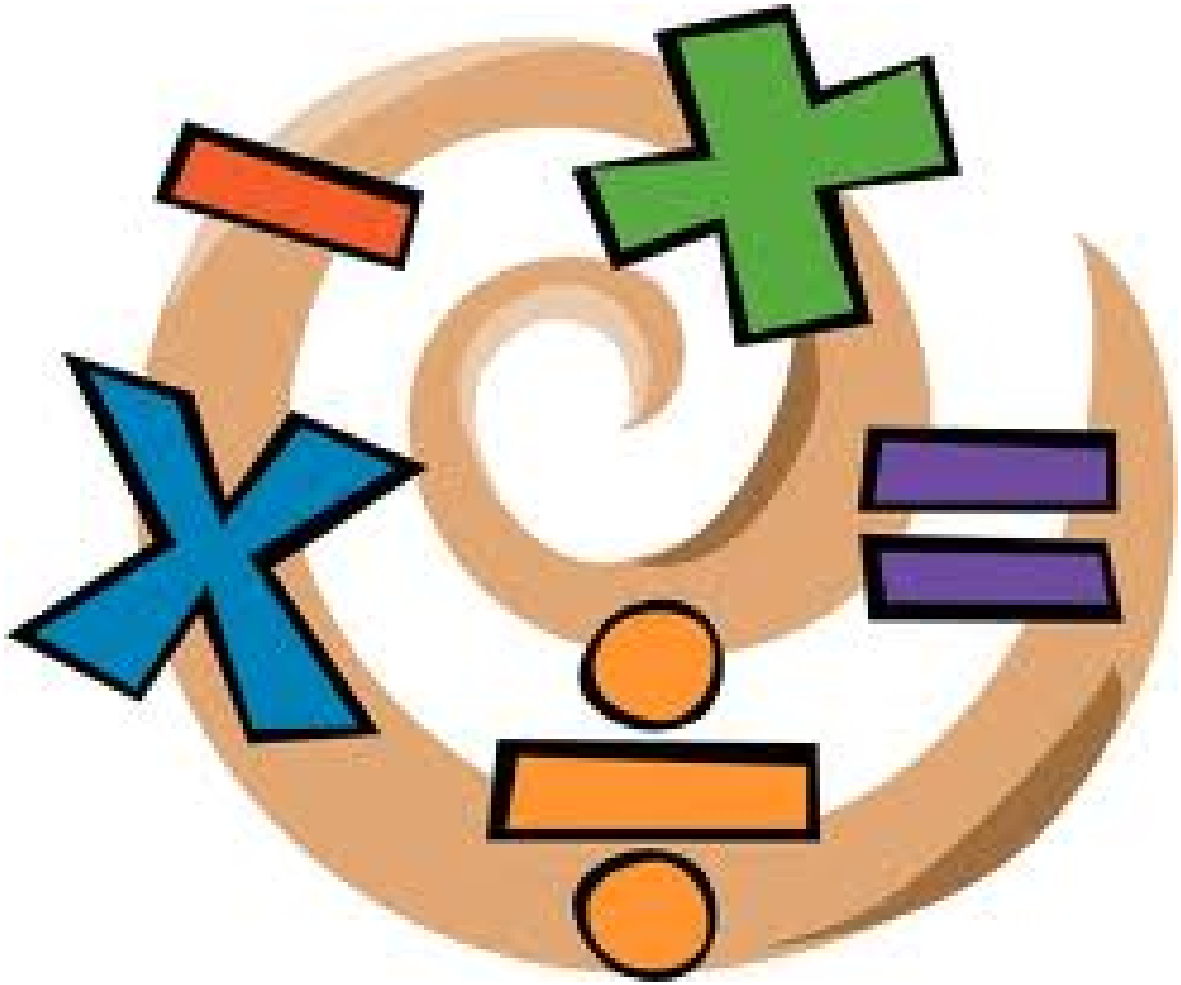
```
public class OutputExample1
{
   public static void main (String [] args)
   {
       int num=100;
       System.out.print(num);
    }
}
```

Introduction to Java: Variable

Java Keywords

abstract	**boolean**	**break**	**byte**	**case**	**catch**	**char**
class	**const**	**continue**	**default**	**do**	**double**	**else**
extends	**final**	**finally**	**float**	**for**	**goto**	**if**
implements	**import**	**instanceof**	**int**	**interface**	**long**	**native**
new	**package**	**private**	**protected**	**public**	**return**	**short**
static	**super**	**switch**	**synchronized**	**this**	**throw**	**throws**
transient	**try**	**void**	**volatile**	**while**		

Introduction to Java: Operator



Common Java Operators / Operator Precedence

Precedence level	Operator	Description	Associativity
1	expression++ expression--	Post-increment Post-decrement	Right to left
2	++expression --expression + - ! ~ (type)	Pre-increment Pre-decrement Unary plus Unary minus Logical negation Bitwise complement Cast	Right to left

Common Java Operators / Operator Precedence

Precedence level	Operator	Description	Associativity
3	* / %	Multiplication Division Remainder/modulus	Left to right
4	+ -	Addition or String concatenation Subtraction	Left to right
5	<< >>	Left bitwise shift Right bitwise shift	Left to right

Common Java Operators / Operator Precedence

Precedence level	Operator	Description	Associativity
6	< <= > >=	Less than Less than, equal to Greater than Greater than, equal to	Left to right
7	= = !=	Equal to Not equal to	Left to right
8	&	Bitwise AND	Left to right
9	^	Bitwise exclusive OR	Left to right

Common Java Operators / Operator Precedence

Precedence level	Operator	Description	Associativity
10	|	Bitwise OR	Left to right
11	&&	Logical AND	Left to right
12	||	Logical OR	Left to right

Common Java Operators / Operator Precedence

Precedence level	Operator	Description	Associativity
13	=	Assignment	Right to left
	+=	Add, assignment	
	-=	Subtract, assignment	
	*=	Multiply, assignment	
	/=	Division, assignment	
	%=	Remainder, assignment	
	&=	Bitwise AND, assignment	
	^=	Bitwise XOR, assignment	
	|=	Bitwise OR, assignment	
	<<=	Left shift, assignment	
	>>=	Right shift, assignment	

Introduction to Java: Operator

```
public class OutputExample1
{
   public static void main (String [] args)
   {
       int w=100;
       int h=10;
       int area=w * h;
       System.out.print(area);
    }
}
```

Introduction to Java:
Type casting

Introduction to Java: Type casting

TargetVariable= (Target_Type) sourceVariable;

```
double a=10;
int b= (int) a;
```

```
int r=10;
double pi=3.14159265358979323846264338327 95;
float c= (float) (2*pi*r);
```

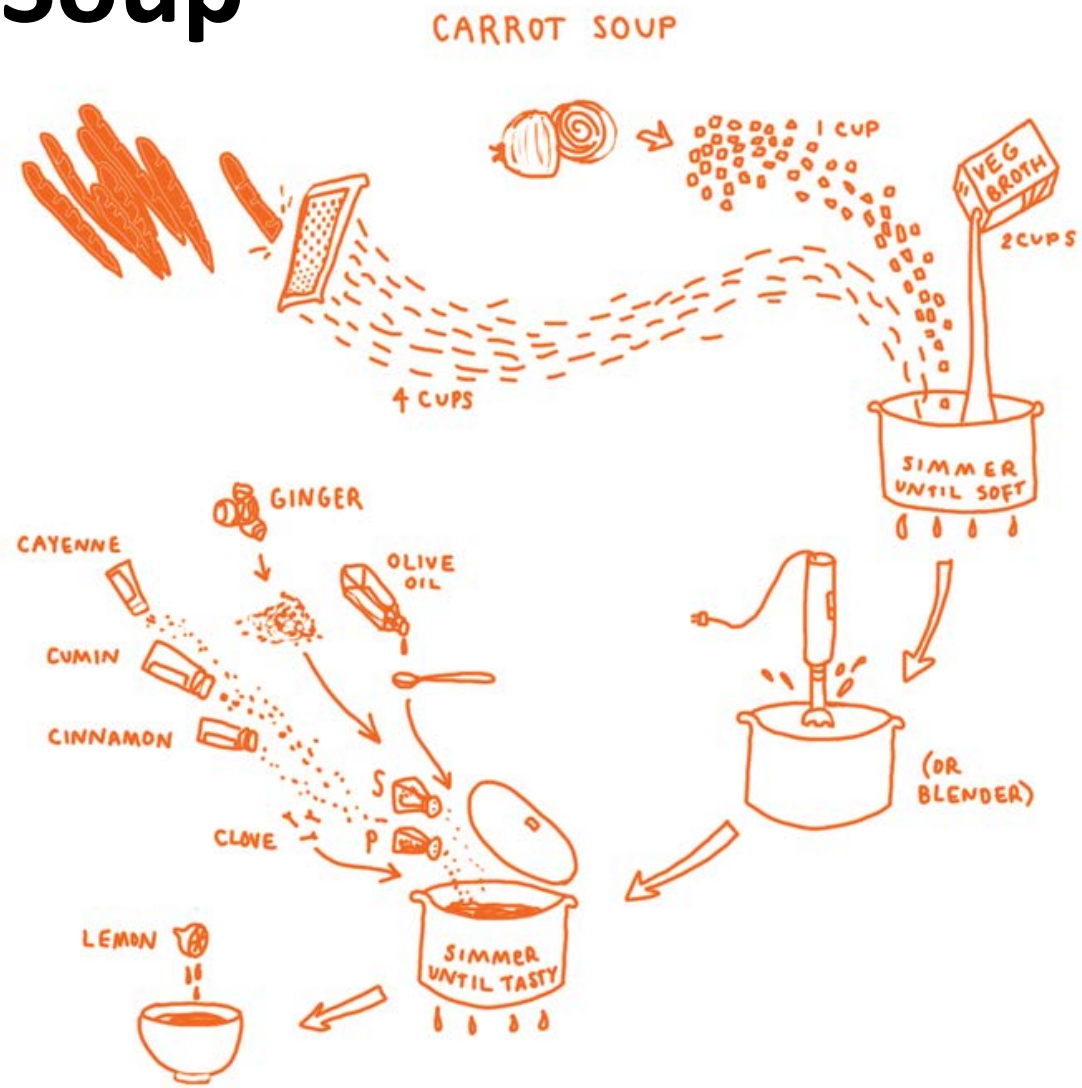
Introduction to Java: Type casting

Introduction to Java: Type casting

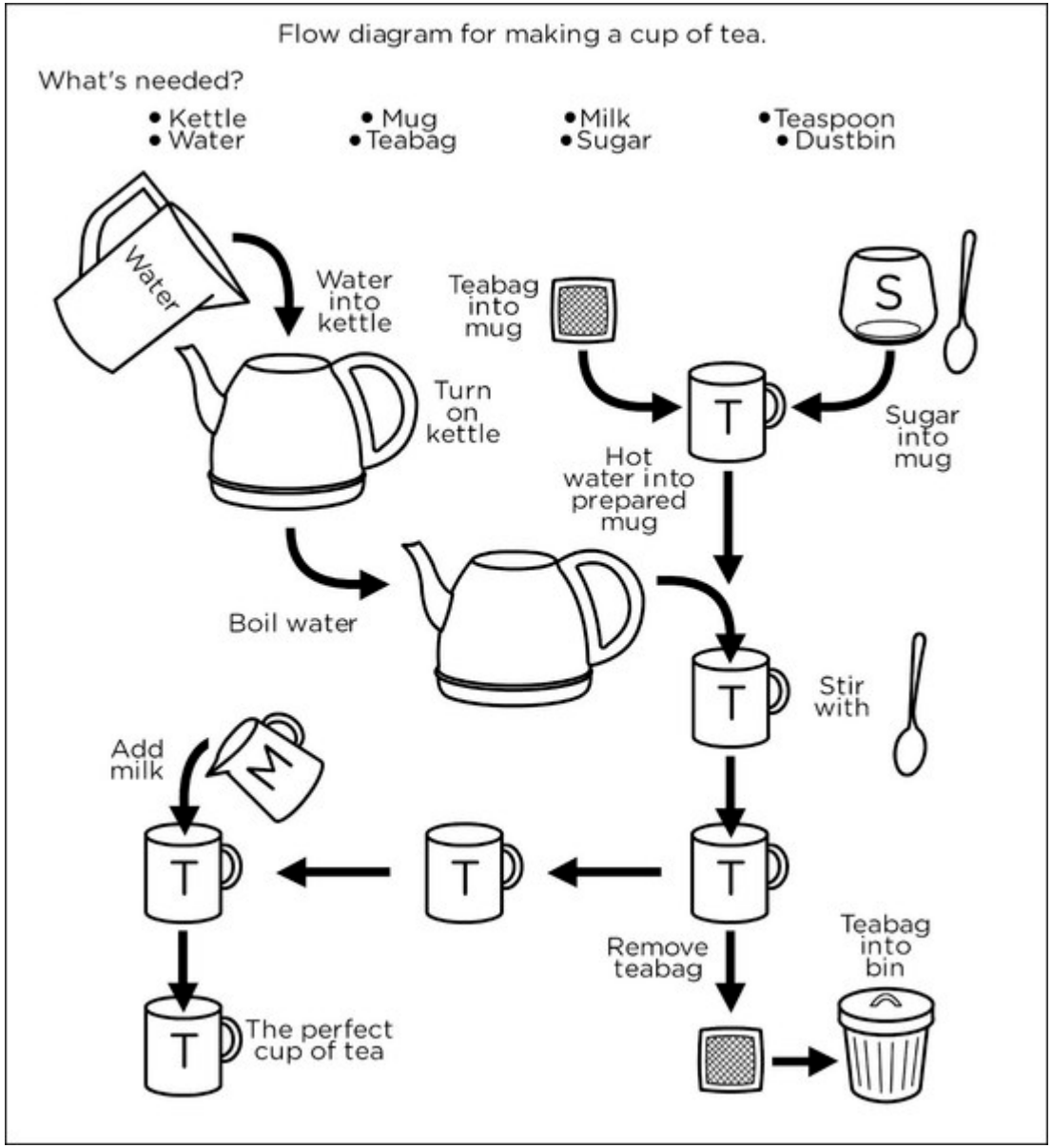
TargetVariable=TargetType.valueof(sourceVariable);

```
String a="10";
int b=Integer.valueOf(a)*2;
```

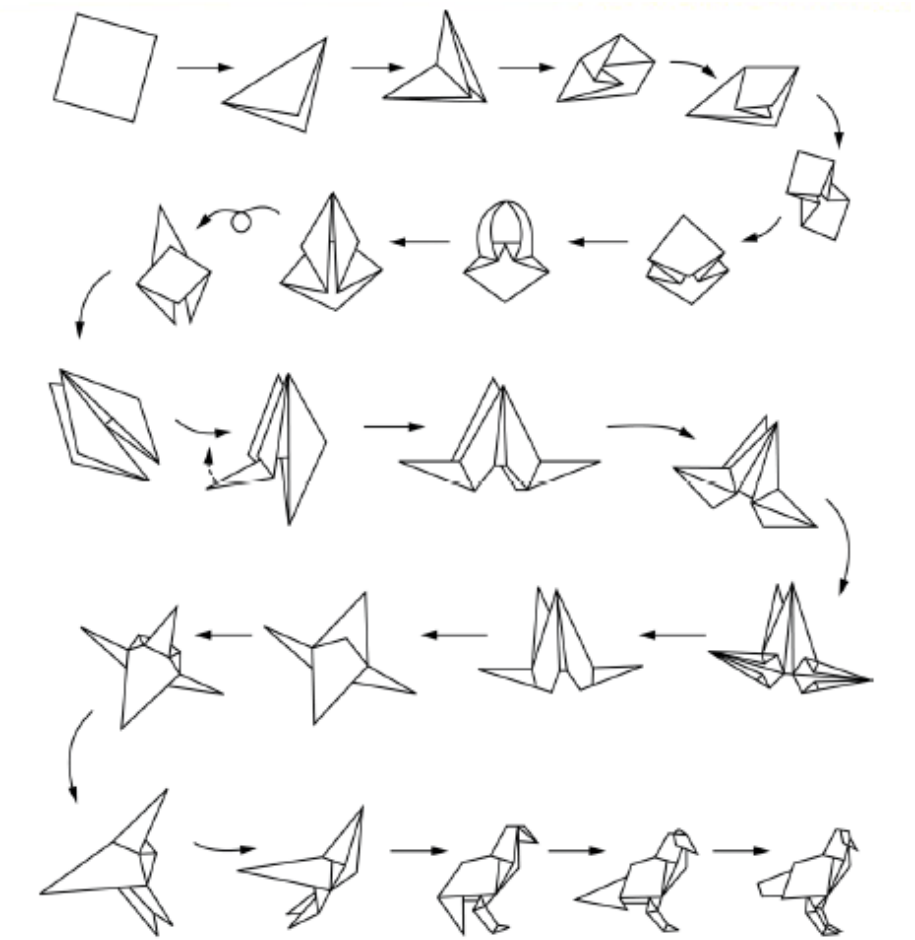
Carrot Soup


CARROT SOUP
1 CUP
VEG BROTH
2CUPS
4 CUPS
SIMMER UNTIL SOFT
GINGER
CAYENNE
OLIVE OIL
CUMIN
CINNAMON
S
P
CLOVE
(OR BLENDER)
LEMON
SIMMER UNTIL TASTY

A cup of tea


Flow diagram for making a cup of tea.
What's needed?
• Kettle
• Water
• Mug
• Teabag
• Milk
• Sugar
• Teaspoon
• Dustbin
Water
Water into kettle
Teabag into mug
S
Turn on kettle
T
Sugar into mug
Hot water into prepared mug
Boil water
T
Stir with
Add milk
M
T
T
T
Remove teabag
Teabag into bin
T
The perfect cup of tea

Origami

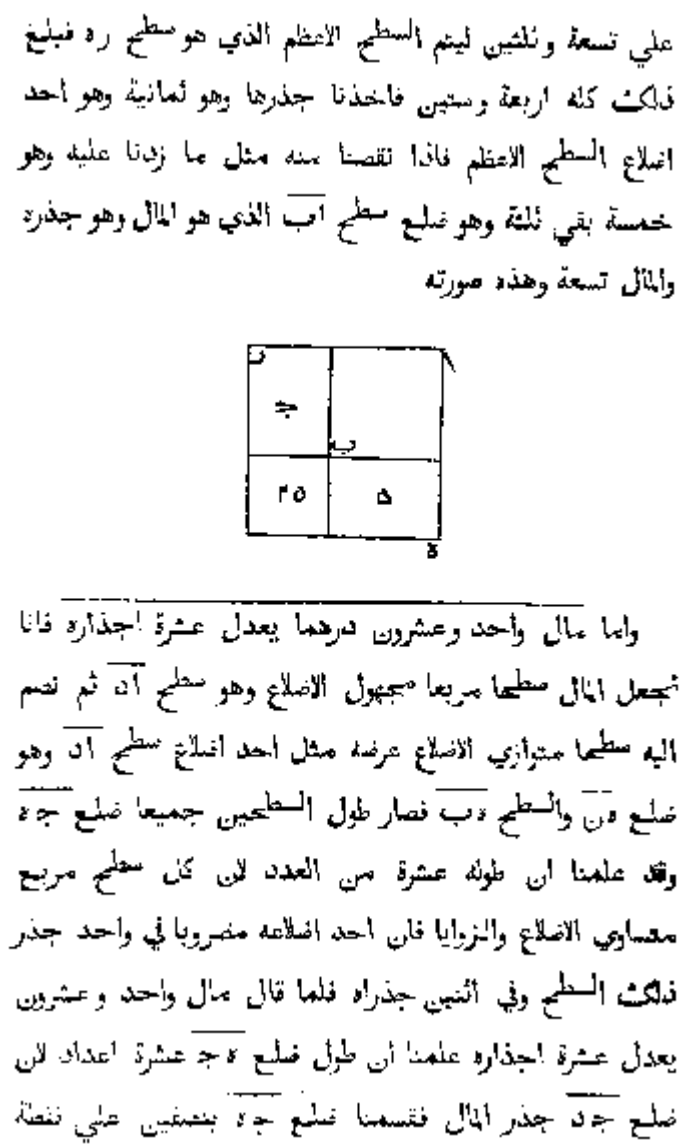


Algorithms

Muhammad ibn Musa al-Khwarizmi

محمد بن موسى الخوارزمى

(c. 780 – c. 850)

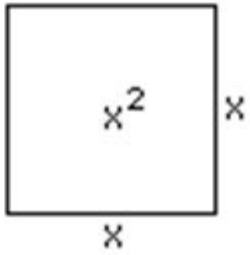


علي تسعة وثلثين ليتم السطح الاعظم الذي هو سطح ره فبلغ
ذلك كله اربعة وستين فاخذنا جذرها وهو ثمانية وهو احد
اضلاع السطح الاعظم فاذا نقصنا منه مثل ما زدنا عليه وهو
خمسة بقي ثلثة وهو ضلع سطح اب الذي هو المال وهو جذره
والمال تسعة وهذه صورته

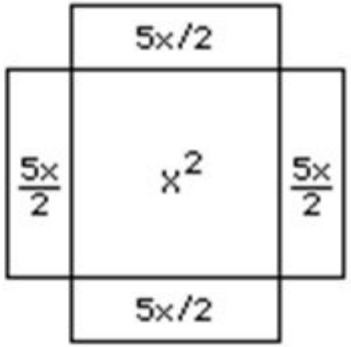
واما مال واحد وعشرون درهما يعدل عشرة اجذاره فانا
نجعل المال سطحا مربعا مجهول الاضلاع وهو سطح اد ثم نضم
اليه سطحا متوازي الاضلاع عرضه مثل احد اضلاع سطح اد وهو
ضلع دن والسطح دب فصار طول السطحين جميعا ضلع جه
وقد علمنا ان طوله عشرة من العدد لان كل سطح مربع
متساوي الاضلاع والزوايا فان احد اضلاعه مضروبا في واحد جذر
ذلك السطح وفي اثنين جذراه فلما قال مال واحد وعشرون
يعدل عشرة اجذاره علمنا ان طول ضلع ه ج عشرة اعداد لان
ضلع جد جذر المال فقسمنا ضلع جه بنصفين علي نقطة

To solve the equation $x^2 + 10x = 39$ by Al-Khwarizmi's "completing the square" method:

Start with a square of side x (which therefore represents x^2).

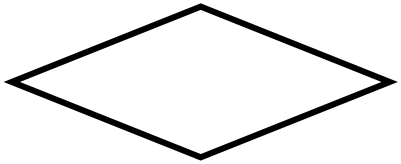


Add to this $10x$ by adding 4 rectangles of length x, and width 10/4. Each small rectangle has an area $10x/4$ (or $5x/2$), total $10x$. We know this has a total area of 39.

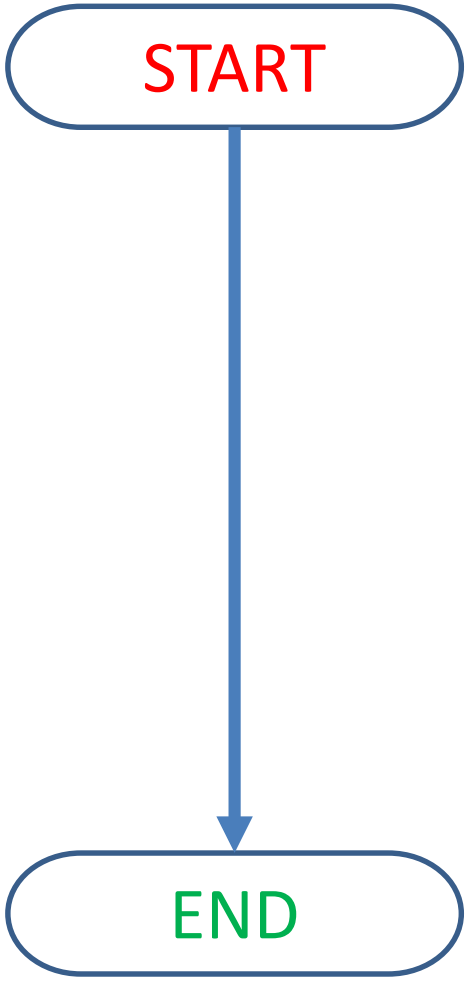


Complete the square by adding 4 little squares with side 5/2 (area of each 25/4). The outside square therefore has an area of 39 + (4 x 25/4) = 39 + 25 = 64. The sides of the outside square are therefore 8. But each side is of length x + 5/2 + 5/2, so x + 5 = 8, giving x = 3.

ผังงาน

สัญลักษณ์	ความหมาย
	จุดเริ่มต้นหรือจุดจบของโปรแกรม (Terminal)
	การรับและแสดงผลของข้อมูล (Input/Output)
	การประมวลผล (Process)
	การตัดสินใจ (Decision /Selection)

ผังงาน



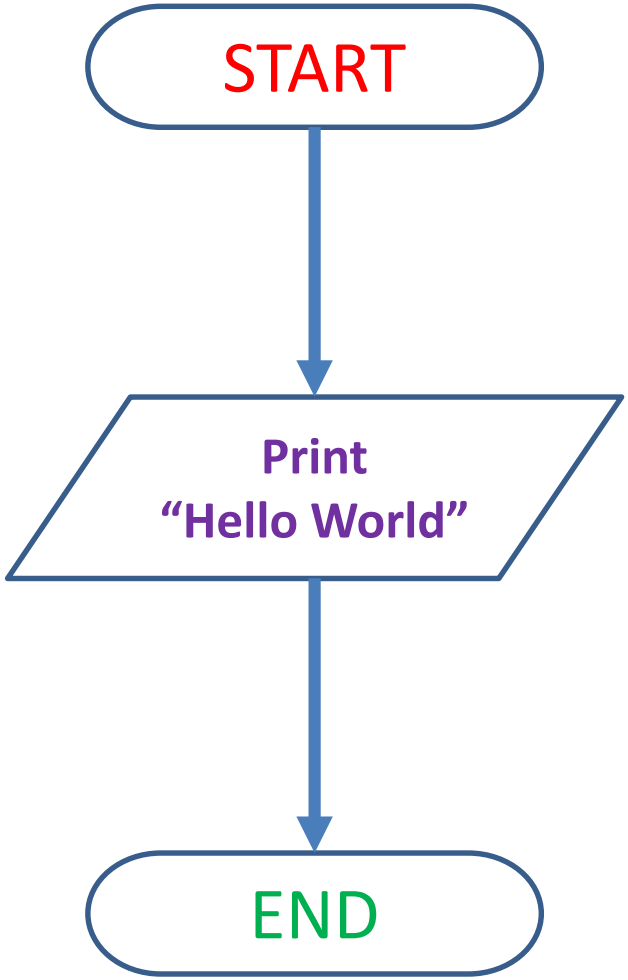
C

```
int main()
{
  return(0);
}
```

Java

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

ผังงาน



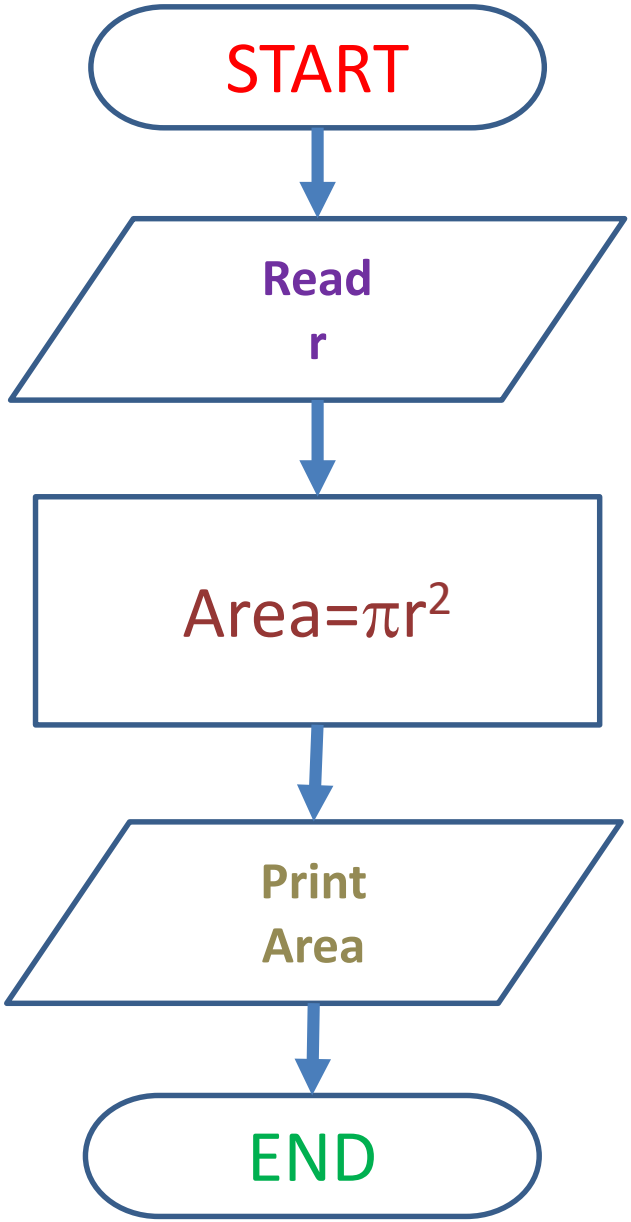
C

```
int main()
{ printf(“Hello World”);
  return(0);
}
```

Java

```
public static void main(String[] args)
 {    System.out.println(“Hello World”);
      System.exit(0);
 }
```

ผังงาน

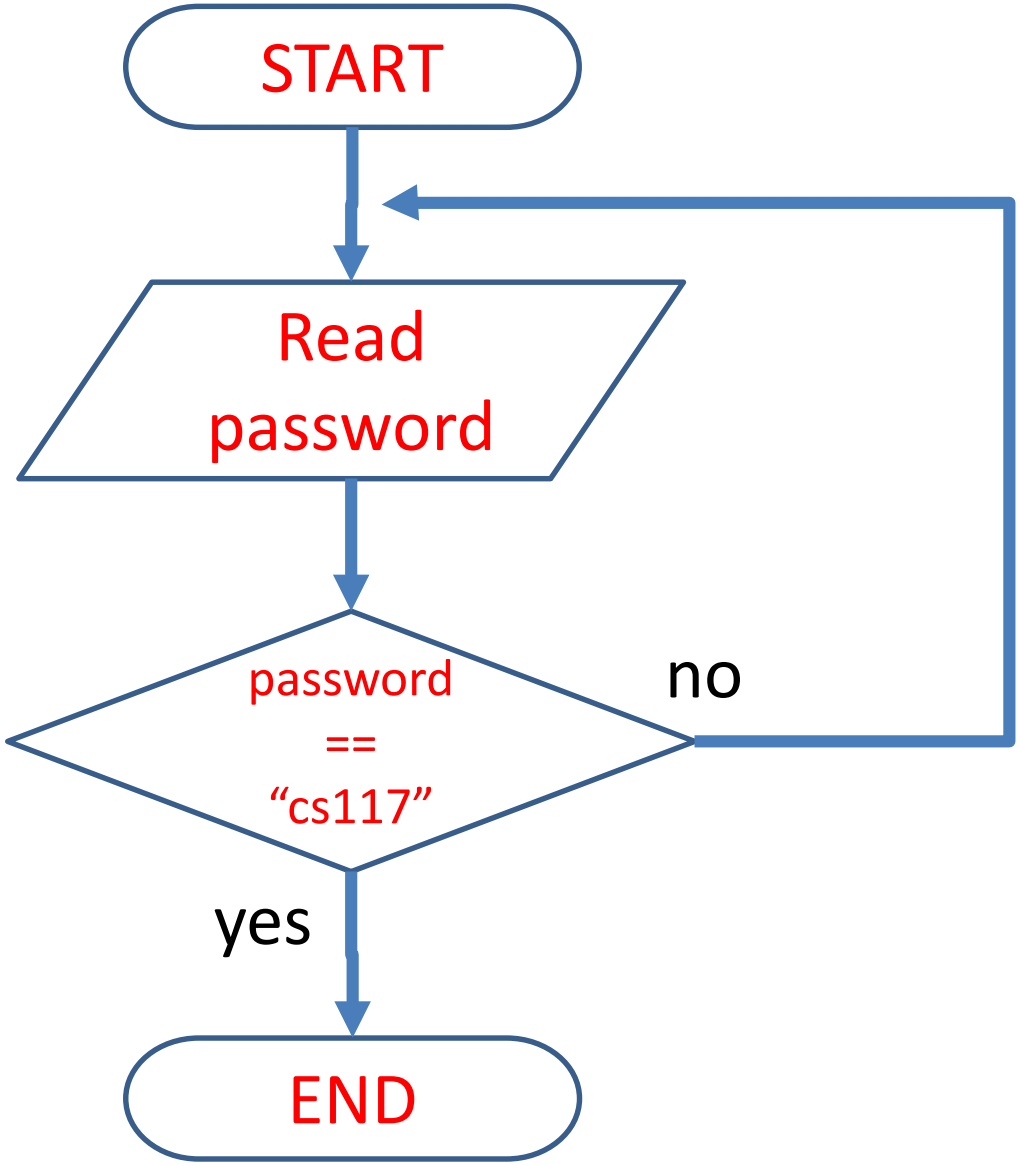


C

```c
int main()
{       double r;
        scanf("%lf",&r);
        double area=r*r*M_PI;
        printf("%lf",area);
    return(0);
}
```

Java

```java
public static void main(String[] args)
 {Scanner s=new Scanner(System.in);
  double r= s.nextDouble();
   double area=r*r*Math.PI;
   System.out.println(area);
     System.exit(0);
 }
```


START
Read
password
password
==
“cs117”
no
yes
END

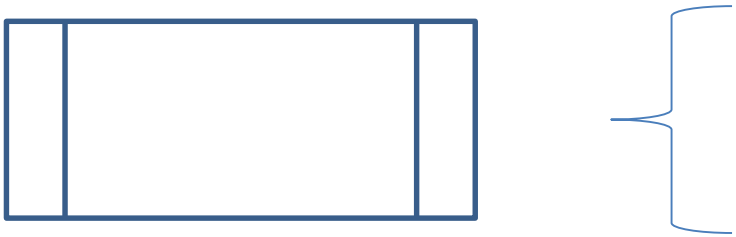
Block Statement

เป็นกลุ่มของชุดคำสั่งที่อยู่ในเครื่องหมายปีกกา **{...}**

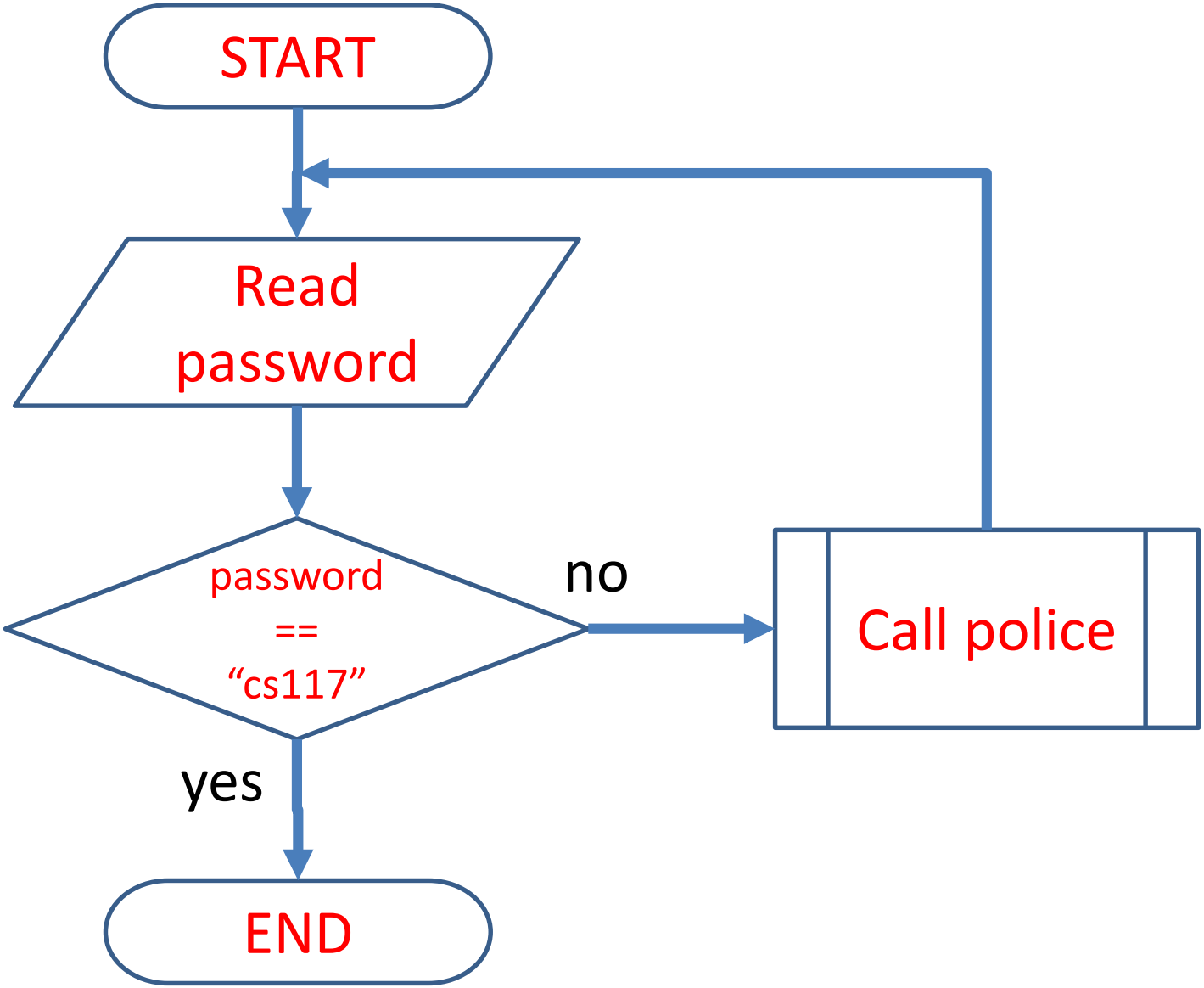
สร้างตัวแปรภายใน Block Statement เรียกว่า **local variable** จะสิ้นสุดการทำงานเมื่อสิ้นสุด block นั้น

```
int a=10;
int b=10;
{
int c=20;
int d=1;
}
int e=20;
{
int f=100;
{
int g=20;
}
int h=1;
}
```

```
int a=10;
int b=10;
 {
  int c=20;
  int d=1;
 }
int e=20;
 {
  int f=100;
    {
      int g=20;
    }
  int h=1;
  }
```



```
int a=10;
int b=10;
 {
  int c=20;
  int d=1;
 }
int e=20;
 {
  int f=100;
    {
      int g=20;
    }
  int h=1;
  }
```


START
Read
password
password
==
“cs117”
no
Call police
yes
END

Class: คลาส

- เป็นพิมพ์เขียว (blueprints) ที่ใช้กำหนดอ๊อบเจ็ค
- **ทุกอ๊อบเจ็คที่มีคุณสมบัติเหมือนกันจะถูกสร้างมาจากคลาสเดียวกัน**
- คลาสประกอบด้วย**ข้อมูลและพฤติกรรม**
- **คลาสต้องมีความเป็นเป็นนามธรรม** (Abstract)

Class = Abstract

Class:คลาส

JAVA	ข้อมูล Properties Attributes	พฤติกรรม Methods Behavior
Car	License plate Make Model Year Color	Start Stop Turn Accelerate Flash lights
Cat	Name Breed Color Age Weight	Sleep Eat Purr Climb Scratch

63

Class:คลาส

สิ่งมีชีวิต
+ ขนาด + อาหาร + สี + การเคลื่อนที่ + การเพิ่มประชากร
+ กิน + เคลื่อนที่ + เพิ่มประชากร + พักผ่อน + ตาย

Properties

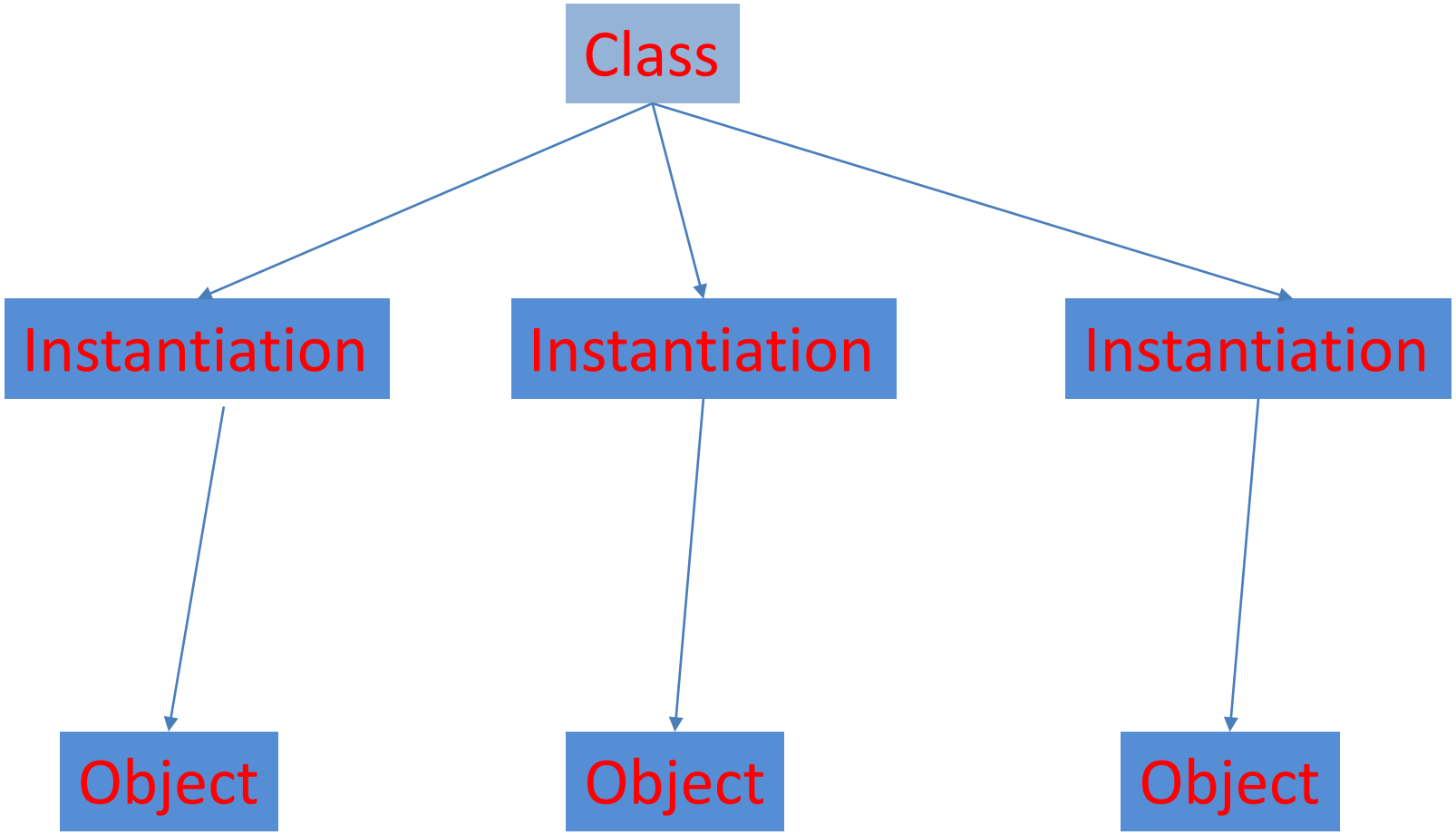
Methods

+ สีของหาง
+ จำนวนของขา

+ กระพริบตา
+ ขยับหาง

64

Object: วัตถุ



65

JAVA เรียก object ว่า instance ของ class

Object: วัตถุ

สิ่งมีชีวิต
+ ขนาด + อาหาร + สี + การเคลื่อนที่ + การเพิ่มประชากร
+ กิน + เคลื่อนที่ + เพิ่มประชากร + พักผ่อน + ตาย

Class

ขนาด: 3x3 เมตร

อาหาร: อ้อย

สี: เทา

การเคลื่อนที่: เดิน 4 ขา

Instantiation

Object Instance

66

Object: วัตถุ

สิ่งมีชีวิต
+ ขนาด + อาหาร + สี + การเคลื่อนที่ + การเพิ่มประชากร
+ กิน + เคลื่อนที่ + เพิ่มประชากร + พักผ่อน + ตาย

Class

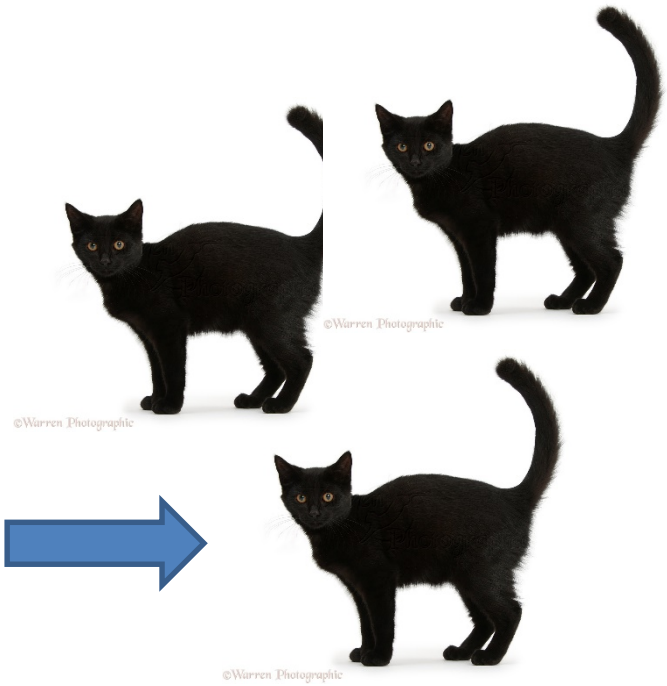
ขนาด: 1 ฟุต

อาหาร: หนู,ปลา,นก

สี: ดำ

การเคลื่อนที่: เดิน 4 ขา

Instantiation



Object Instance

Object: วัตถุ

สิ่งมีชีวิต
+ ขนาด + อาหาร + สี + การเคลื่อนที่ + การเพิ่มประชากร
+ กิน + เคลื่อนที่ + เพิ่มประชากร + พักผ่อน + ตาย

Class

ขนาด: 4um

อาหาร:น้ำตาล

สี: เหลือง

การเพิ่มประชากร: แบ่งตัว

Instantiation

Object Instance

68

Object: วัตถุ

สิ่งมีชีวิต
+ ขนาด + อาหาร + สี + การเคลื่อนที่ + การเพิ่มประชากร
+ กิน + เคลื่อนที่ + เพิ่มประชากร + พักผ่อน + ตาย

Class

Object

Object . กิน (หนู)

วัตถุ Method Input parameter

Object: วัตถุ

สิ่งมีชีวิต
+ ขนาด + อาหาร + สี + การเคลื่อนที่ + การเพิ่มประชากร
+ กิน + เคลื่อนที่ + เพิ่มประชากร + พักผ่อน + ตาย

Class

Object

Object2 = Object . เพิ่มประชากร ( )

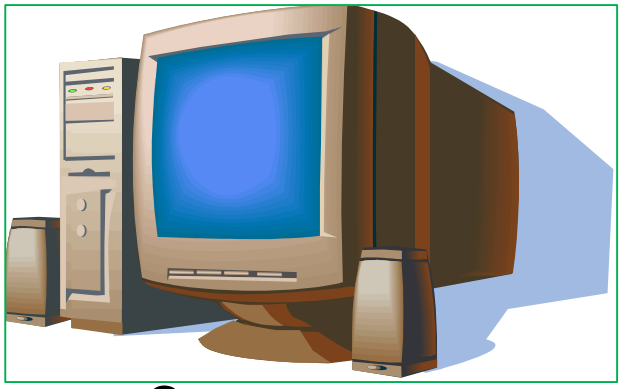
Return parameter

วัตถุ

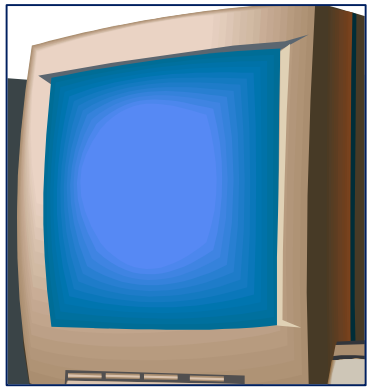
Method

Introduction to Java: Object, Methods, Properties

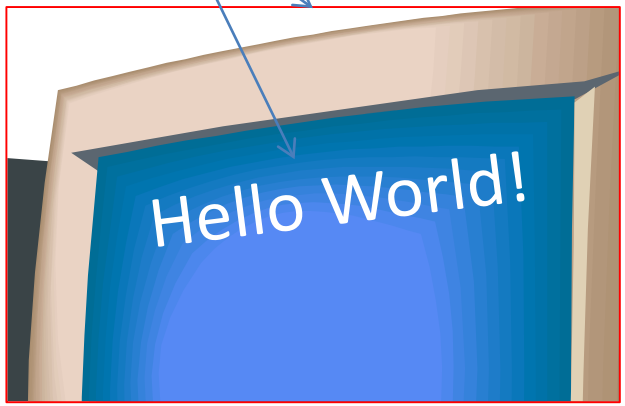
```
System.out.println("Hello World!");
```



System object



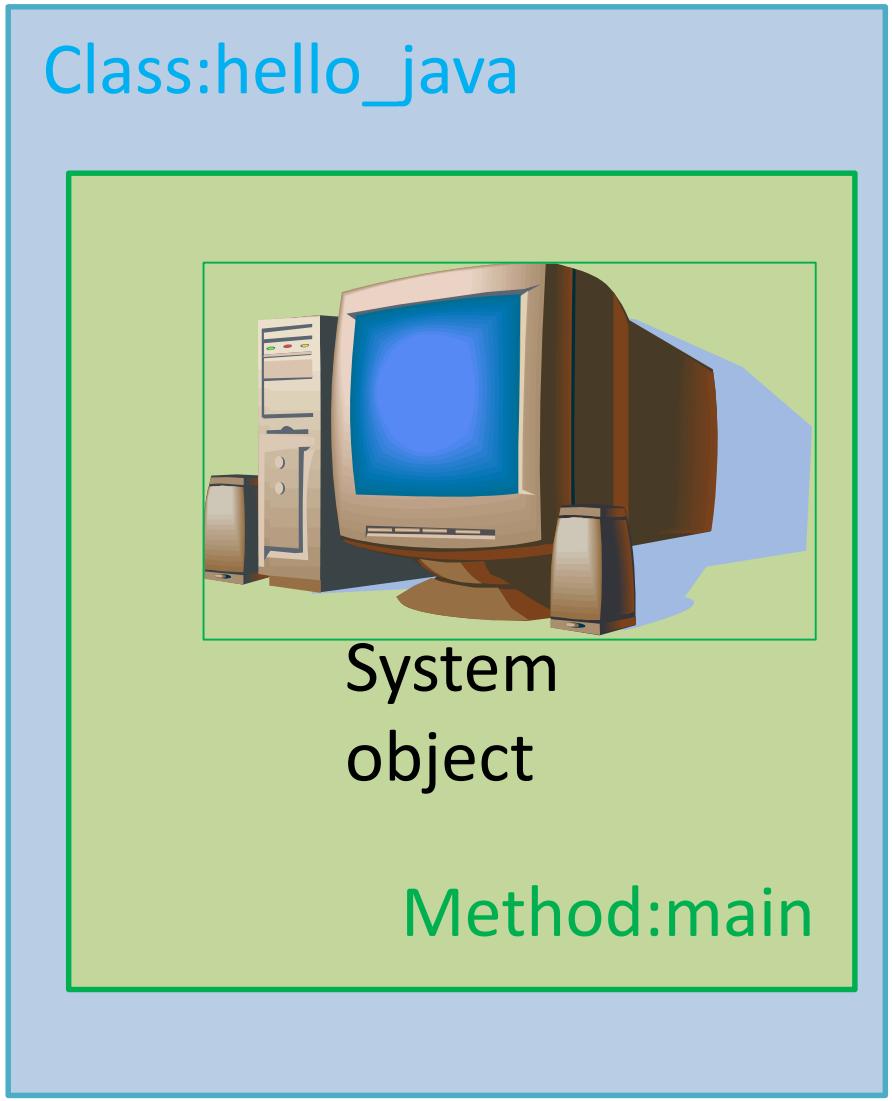
Output object



Print 1 Line method

Introduction to Java: Object, Methods, Properties

```
public class hello_java {

public static void main(String[] args) {
System.out.println("Hello world!");
}
}
```



Introduction to Java:
Object, Methods, Properties

System.in.*

SCANNER

SCANNER
+ + + +
+nextInt () +nextLong () +nextFloat () +nextDouble () +nextLine ();

class

Input: System.in

Instantiation

Instance

Introduction to Java: Read keyboard input

```
import java.util.Scanner;
public class hello_java {
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
String str= sc.next();
System.out.println(str);
}
}
```

Introduction to Java: Read keyboard input

```
import java.util.Scanner;
public class hello_java {
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
String str= sc.next();
System.out.println(str);
}
}
```

Input parameter

Class

Instance (object)

Instantiation

Introduction to Java: Read keyboard input

```
import java.util.Scanner;
public class hello_java {
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
String str= sc.next();
System.out.println(str);
}
}
```

Method

Introduction to Java: Read keyboard input

Useful Methods Of Class Scanner

```
nextInt ()
nextLong ()
nextFloat ()
nextDouble ()
nextLine ();
```

Introduction to Java: Read keyboard input

```
import java.util.Scanner;
public class hello_java {
public static void main(String[] args) {
Scanner sc;
System.out.print("Please input height:");
sc=new Scanner(System.in);
int h=sc.nextInt();
System.out.print("Please input height:");
sc.nextLine();
int w=sc.nextInt();
System.out.println("area="+0.5*h*w);
}
}
```

Introduction to Java:Math class

abs()	ใช้หาค่าสัมบูรณ์
sqrt()	ใช้หาค่ารากที่สอง
pow()	ใช้หาค่ายกกำลัง
ceil()	ใช้ปัดเศษทศนิยมให้มีค่ามากขึ้น
floor()	ใช้ปัดเศษทศนิยมให้มีค่าน้อยลง
round()	ใช้ปัดเศษทศนิยมตามปกติ (คือ มากกว่าหรือเท่ากับ 5 จะปัดขึ้น และ น้อยกว่า 5 จะปัดลง)
random()	ใช้ในการสุ่มตัวเลข ระหว่าง 0 ถึง 1
max()	ใช้ในการหาค่าที่มากกว่าระหว่างตัวเลขที่กำหนด
min()	ใช้ในการหาค่าที่น้อยกว่าระหว่างตัวเลขที่กำหนด
log()	ใช้หาค่า **log**
exp()	ใช้หาค่า **e** ยกกำลัง (**e** มีค่าประมาณ 2.71828)
sin()	ใช้หาค่า **sine** ของมุมที่กำหนด
cos()	ใช้หาค่า **cosine** ของมุมที่กำหนด
tan()	ใช้หาค่า **tan** ของมุมที่กำหนด
asin()	ใช้หาค่า **arc sine** ของตัวเลขที่กำหนด
acos()	ใช้หาค่า **arc cosine** ของตัวเลขที่กำหนด
atan()	ใช้หาค่า **arc tangent** ของตัวเลขที่กำหนด
atan2()	ใช้หาค่ามุมพิกัดโพลาร์ ของพิกัด **x** และ **y**
PI	ใช้คืนค่า **PI** ซึ่งเท่ากับ 3.141592653589793

Introduction to Java: Math class

$$c = \sqrt{a^b}$$

```
C =Math.sqrt(Math.pow(a,b));
```

Introduction to Java: Math class

Assignment

-Area of circle calculator
-Currency converter
-Temperature converter

Introduction to Java: String Formatter

type character	input	string result
%d	signed int	signed decimal integer
%u	unsigned int	unsigned decimal integer
%o	unsigned int	unsigned octal integer
%x, %X	unsigned int	unsigned hexadecimal integer, lowercase or uppercase
%z[*n*], %Z[*n*]	unsigned int	unsigned integer base *n*, with *n* coded in decimal; include square brackets
%f	float	real number, standard notation
%e, %E	float	real number, scientific notation (lowercase or uppercase exponent marker)
%g, %G	float	same format as %f or %e, depending on the value. Scientific notation is used only if the exponent is greater than the precision or less than -4.
%s	String	string
%c	char	character
%p	Object	object identity hash code (i.e., pointer value), in unsigned hexadecimal

Introduction to Java: String Formatter

```
System.out.println(String.format("%f", Math.PI));
```

```
3.141593
```

```
System.out.println(String.format("%.10f", Math.PI));
```

```
3.1415926536
```

```
System.out.println(String.format("%.2f", Math.PI));
```

```
3.14
```

การบ้าน 2

นาฬิกา
+ + + + +
+ + + + + +

ให้ออกแบบคลาสนาฬิกา พร้อมทั้งยกตัวอย่างการนำเอาคลาสที่ออกแบบไปสร้างเป็นนาฬิกา 3 ชนิด คือ

1 นาฬิกาผนัง

2 นาฬิกาข้อมือ

3 นาฬิกาปลุก

จำนวน properties และ method ให้ใส่ไปตามความเหมาะสม

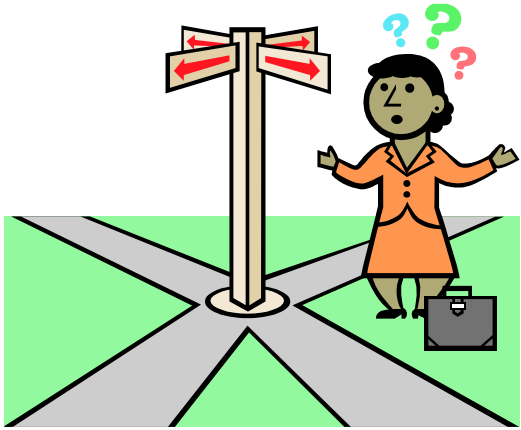
เขียนคำตอบใส่กระดาษ A4 ส่งทางกล่องรับการบ้าน หน้าแลป 2

เขียนชื่อ รหัส กลุ่มเรียน ที่<u>มุมบนขวา</u>

ภายในวันศุกร์ที่ 14 กรกฎาคม 2560

Introduction to Java: Decision making

```
If(Boolean Expression)
{
        do something if true
}else
{
        Do something if not true
}
```



Introduction to Java: Decision making

```
If( (a/2) == 1)
{
        System.out.println("a is event number");
}else
{
        System.out.println("a is odd number");
}
```

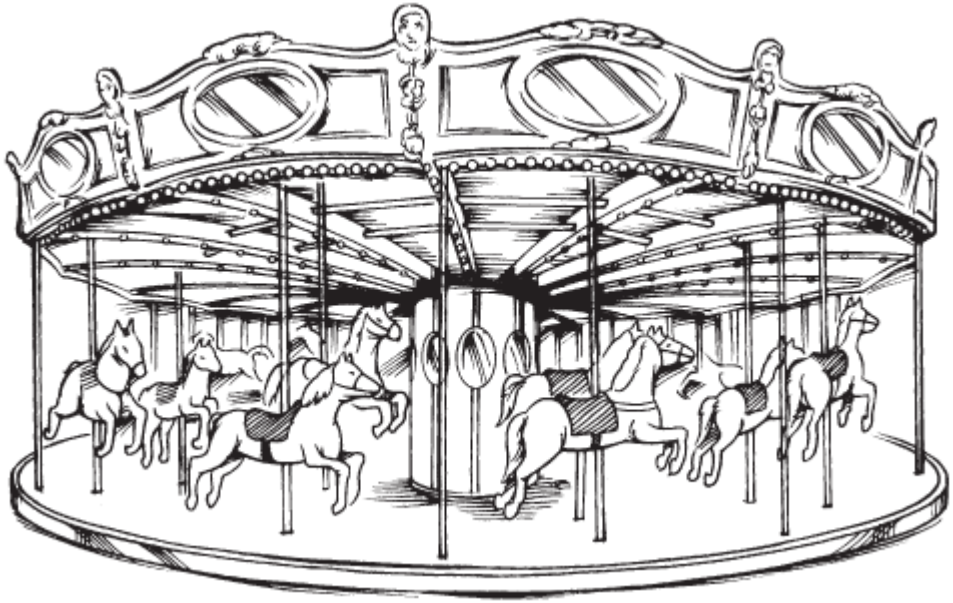
Introduction to Java: Decision making If,Else-if

```
if (Boolean expression)
{
Body of if
        }else if (Boolean expression)
    {
Body of first else-if
                :       :    :
        }else if (Boolean expression)
    {
Body of last else-if
        }else
          {
 Body of else}
```

Introduction to Java: Decision making If,Else-if

```
 if (gpa == 4)
{
        System.out.println("Excellent");
  } else if (gpa == 3)
     {
       System.out.println("Good");
     } else if (gpa == 2)
        {
           System.out.println("Fair");
         }
           else
            {
                System.out.println("Fail!!");
            }
```

Introduction to Java: Loop



Introduction to Java: Loop

Java Pre-test loops

- For
- While

Java Post-test loop

- Do-while

Introduction to Java: While

Format:

```
while (Boolean expression)
    Body
```

Example:

```
int i = 1;
while (i <= 10)
{

    System.out.println(i);
    i = i + 1;
}
```

```
1
2
3
4
5
6
7
8
9
10
```

Introduction to Java: for

Format:

```
for (initialization; Boolean expression; update control)
        Body
```

Example:

```
for( int i=1; i<=10; i++ )
{
 System.out.println(i);
 }
```

```
1
2
3
4
5
6
7
8
9
10
```

Introduction to Java: do-while

Format:

```
do
    Body
while (Boolean expression);
```

Example:

```
int i=1;
do{
    System.out.println(i);
    i++;
}while( i<=10 );
```

```
1
2
3
4
5
6
7
8
9
10
```

Introduction to Java: do-while

Format:

```
do
    Body
while (Boolean expression);
```

Example:

```
int i=1;
do{
    System.out.println(i);
    i++;
}while( i<=0 );
```

1

Introduction to Java: loop

```
for(int i=1;i<20;i++)
{
  System.out.println(String.format("%d x 12 = %d", i,i*12));
}
```

```
1 x 12 = 12
2 x 12 = 24
3 x 12 = 36
4 x 12 = 48
5 x 12 = 60
6 x 12 = 72
7 x 12 = 84
8 x 12 = 96
9 x 12 = 108
10 x 12 = 120
11 x 12 = 132
12 x 12 = 144
13 x 12 = 156
14 x 12 = 168
15 x 12 = 180
16 x 12 = 192
17 x 12 = 204
18 x 12 = 216
19 x 12 = 228
```

Introduction to Java: loop

```
int h=10;
for(int y=1;y<=h;y++)
{
  for(int x=1;x<=h;x++)
  {
     System.out.print("*");
  }
     System.out.println("");
}
```

```
**********

**********

**********

**********

**********

**********

**********

**********

**********

**********
```

Introduction to Java: loop

```
*

**

***

****

*****

******

*******

********

*********

**********
```

Problems | Javadoc | Declaration | Console

```
<terminated> hello_java [Java Application] C:\Program Files\Ja
$@@@@@@@@@@
$$@@@@@@@@@
$$$@@@@@@@@
$$$$@@@@@@@
$$$$$@@@@@@
$$$$$$@@@@@
$$$$$$$@@@@
$$$$$$$$@@@
$$$$$$$$$@@
$$$$$$$$$$@
$$$$$$$$$$$
```

Introduction to Java: Array and Vector

Introduction to Java: Array

Format

type variable[]=new type[size];

Example

int A[]=new int[10];

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]

A[4]=100;

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
				100					

Introduction to Java: Array

Example

```
int A[][]=new int[3][3];
```

A[0][0]	A[0][1]	A[0][2]
A[1][0]	A[1][1]	A[1][2]
A[2][0]	A[2][1]	A[2][2]

Introduction to Java: Array

```
int a[]=new int[10];
 for(int i=0;i<10;i++)
 {
          System.out.print(String.format("Input %d:", i));
            Scanner sc=new Scanner(System.in);
            a[i]=sc.nextInt();
 }

  Arrays.sort(a);

 for(int i=0;i<10;i++)
 {
         System.out.print(String.format(" %d,", a[i]  ));
 }
```

Introduction to Java: Array

Introduction to Java: Vector

Format

```
Vector<Object> vector=new Vector<Object>();
vector.add(value);
```

Example

```
Vector <Integer> A=new Vector<Integer>();
A.add(2);
A.add(8);
```

Introduction to Java: Vector

```
Vector<Integer> a=new Vector<Integer>();
int i;
  do{
  System.out.print("Please input number (type 0 when done):");
  Scanner sc=new Scanner(System.in);
   i=sc.nextInt();
   a.add(i);
  }while(i !=0);
System.out.println( a.toString() );
```

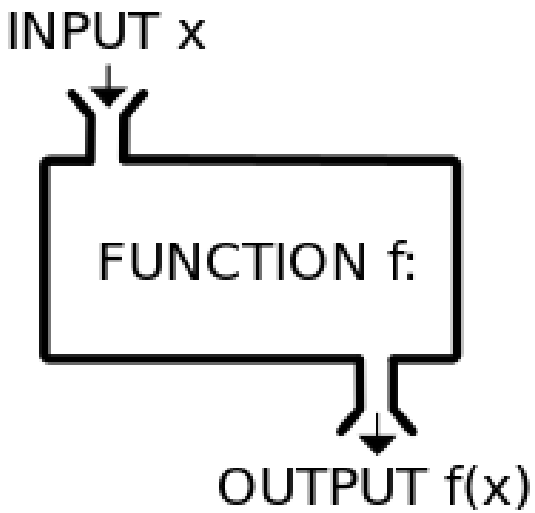
Introduction to Java: Vector

```
Integer b[]=new Integer[ a.size( ) ];
    a.toArray(b);
    Arrays.sort(b);
System.out.println( a.toString() );
  for(int j=0;j<b.length;j++)
          {
          System.out.print(String.format(" %d,", b[j]  ));
          }
```

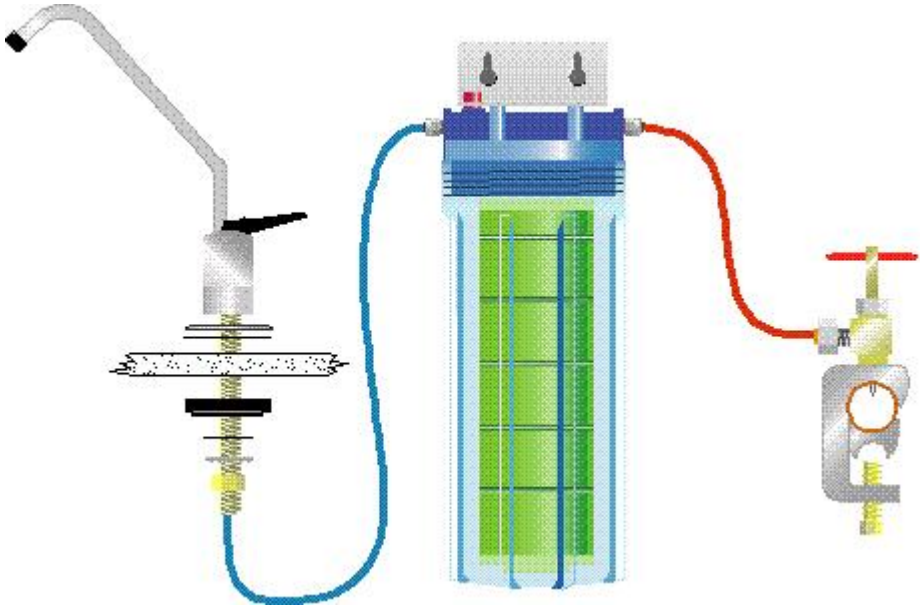
Function, Subroutine and Class

Introduction to Java: function

A function is a relation between a set of inputs and a set of permissible outputs with the property that each input is related to exactly one output.



Introduction to Java: function



Introduction to Java: function

Format

```
private static ReturnType fnName(Type params)
{
        return ReturnValue;
}
```

Example

```
private static int Adder( int x , int y )
{
        return x+y;
}
```

Introduction to Java: Subroutine

Format

```
private static void fnName(Type params)
{

}
```

Example

```
private static void Printer( int x)
{
 System.out.println(String.format("^_^ %d",x));
}
```

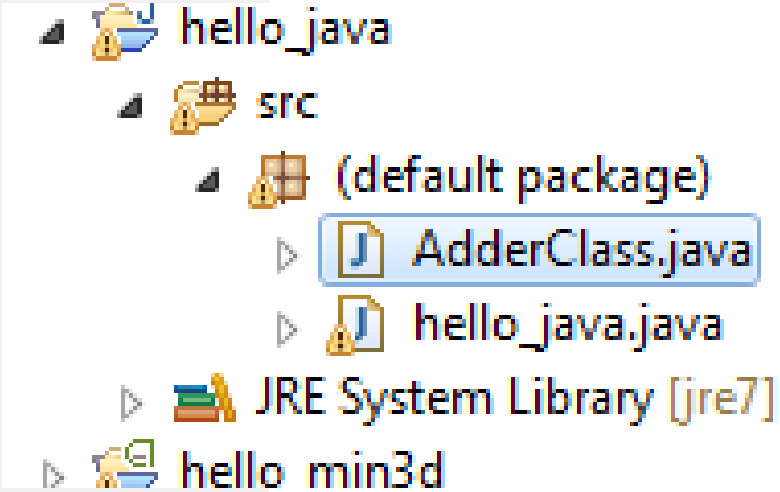
Introduction to Java: Class

```
public class MyClass {
  // method

}
```

Introduction to Java: Class

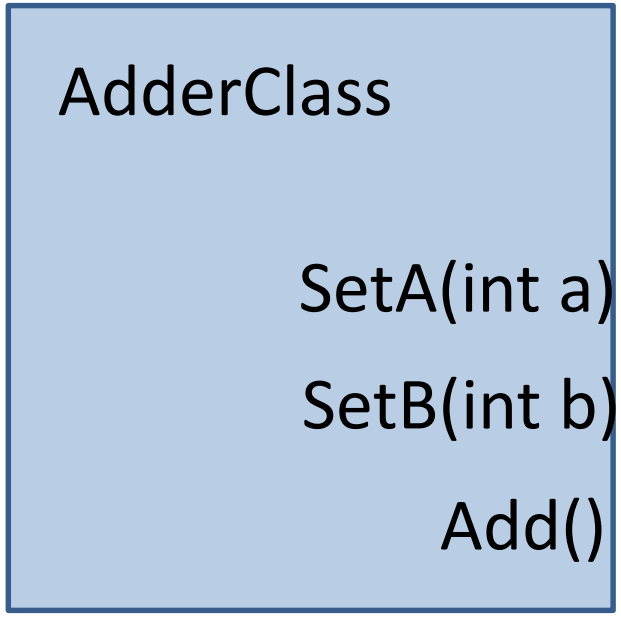
Class: AdderClass

Method:SetA,SetB,Add

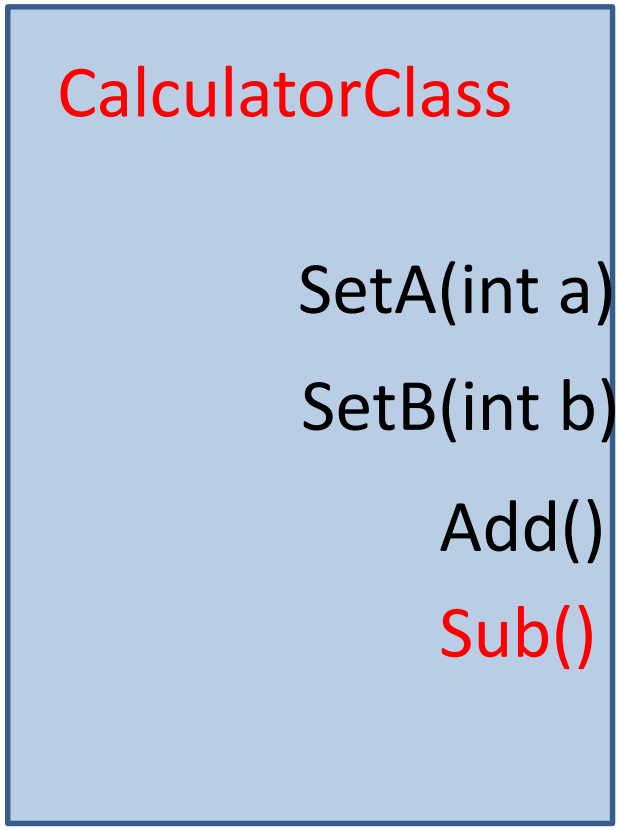


Introduction to Java: Class Instance

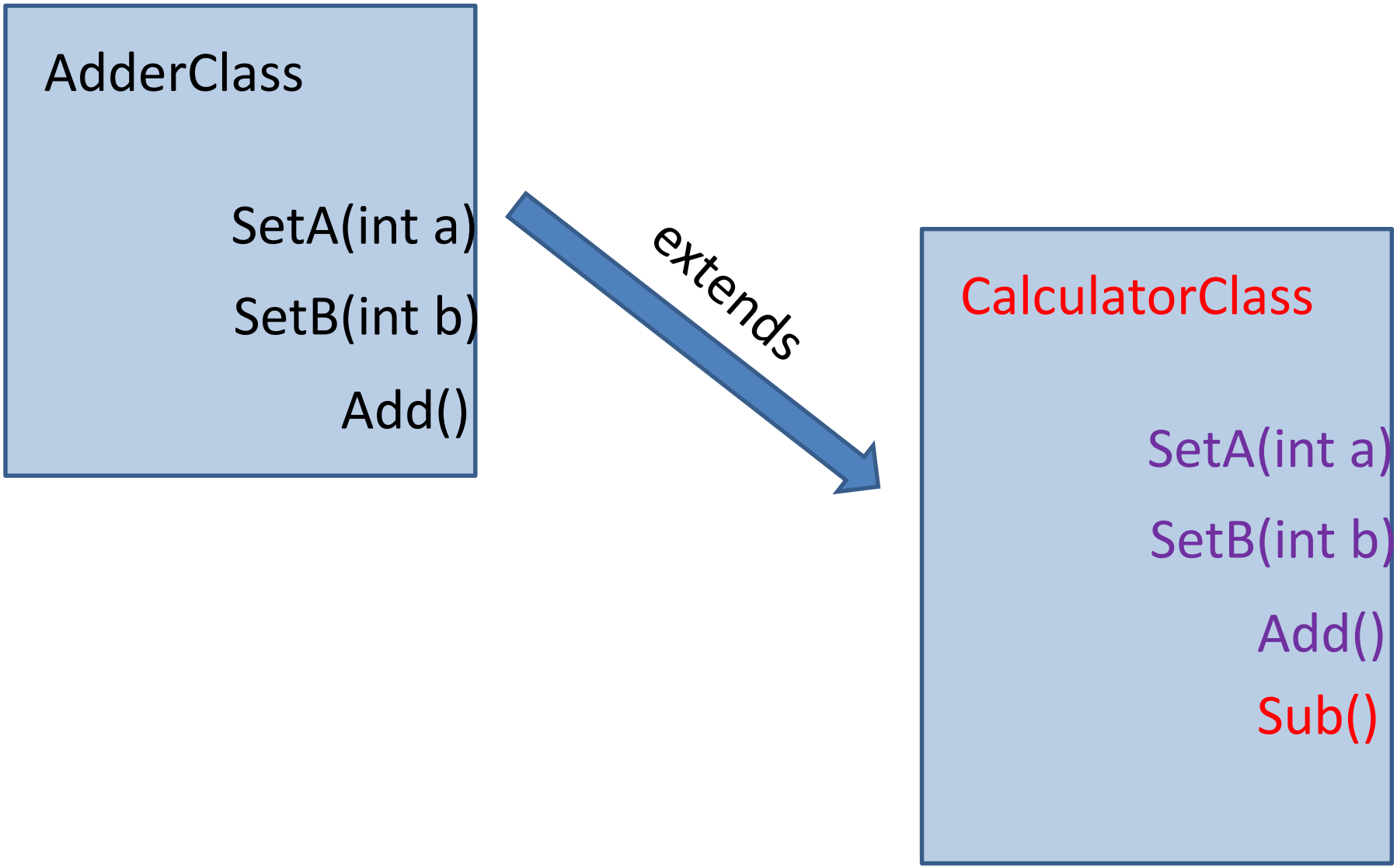
```
AdderClass adderclass=new AdderClass();
adderclass.SetA(10);
adderclass.SetB(20);
System.out.print(adderclass.Add());
```



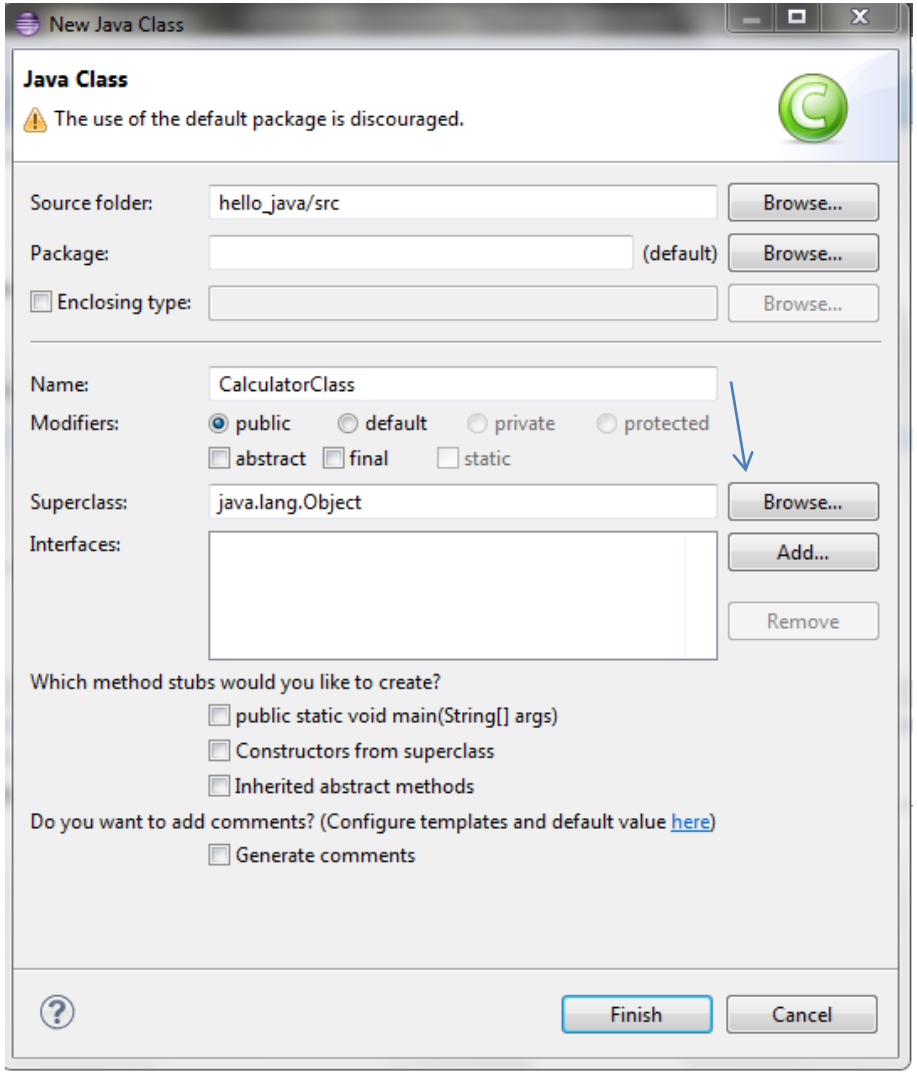
Introduction to Java: Class Inheritance

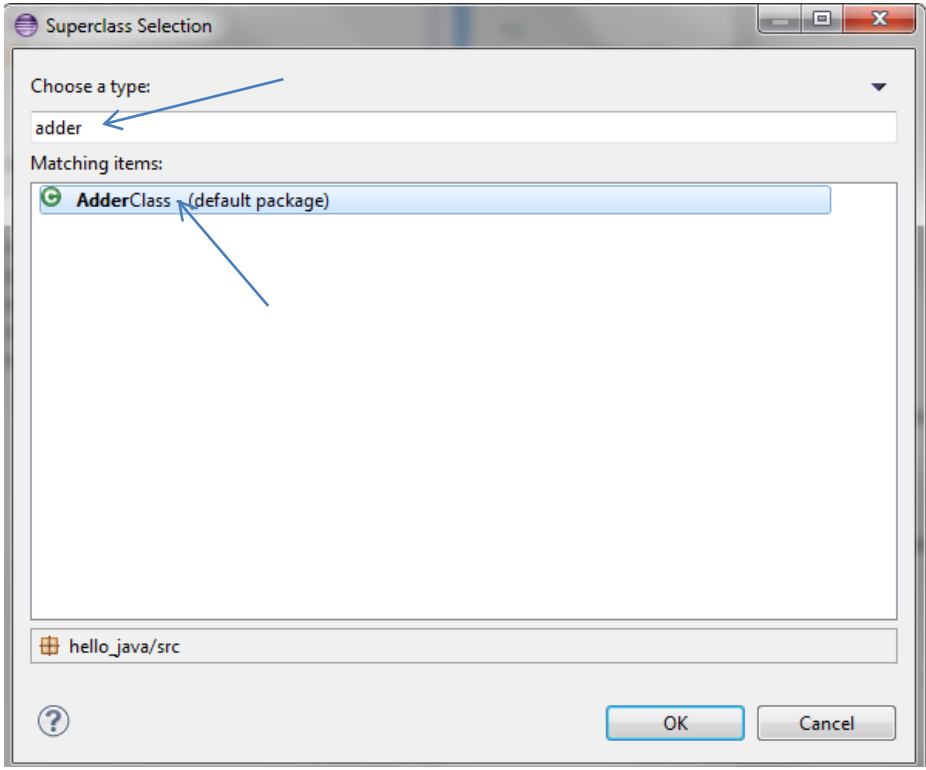


Introduction to Java: Class Inheritance

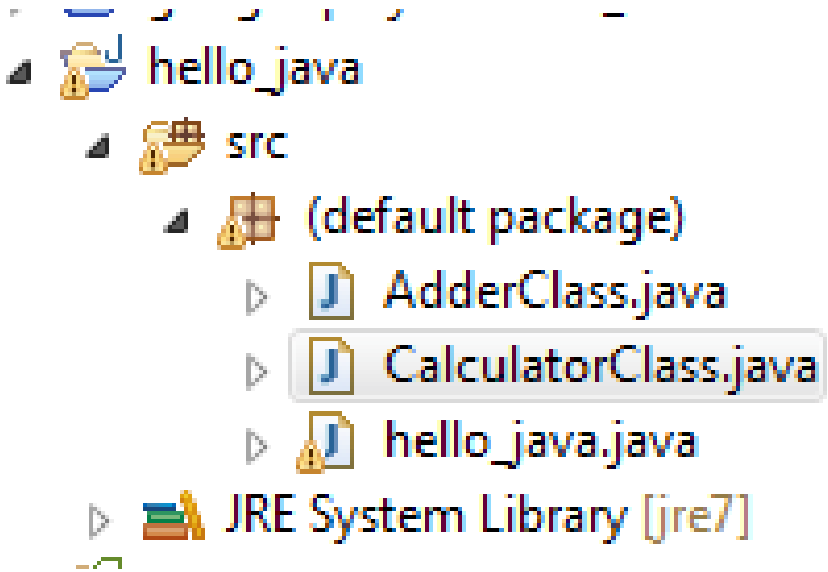


Introduction to Java: Class Inheritance





Introduction to Java: Class Inheritance




```
public class CalculatorClass extends AdderClass {
public int Sub(){
        return A-B;
}
}
```

Introduction to Java: Class Inheritance

```
CalculatorClass calculator=new CalculatorClass();
calculator.SetA(10);
calculator.SetB(20);
System.out.print(calculator.Add());
System.out.print(calculator.Sub());
```

Introduction to Java: Method Overide

CalculatorClass

SetA(int a)

SetB(int b)

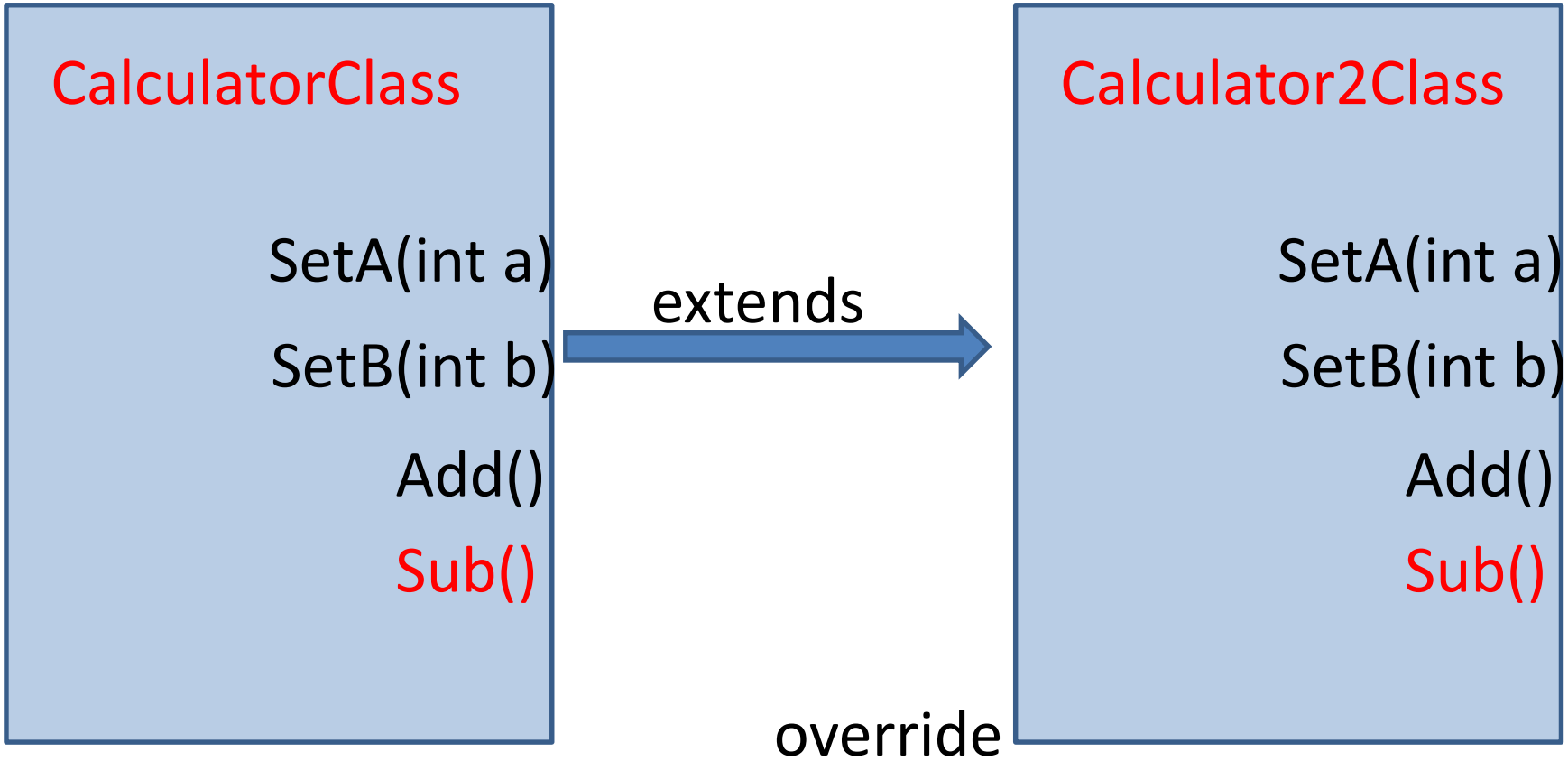
Add()

Sub()

```
public int Sub(){
        return A-B;
}
```

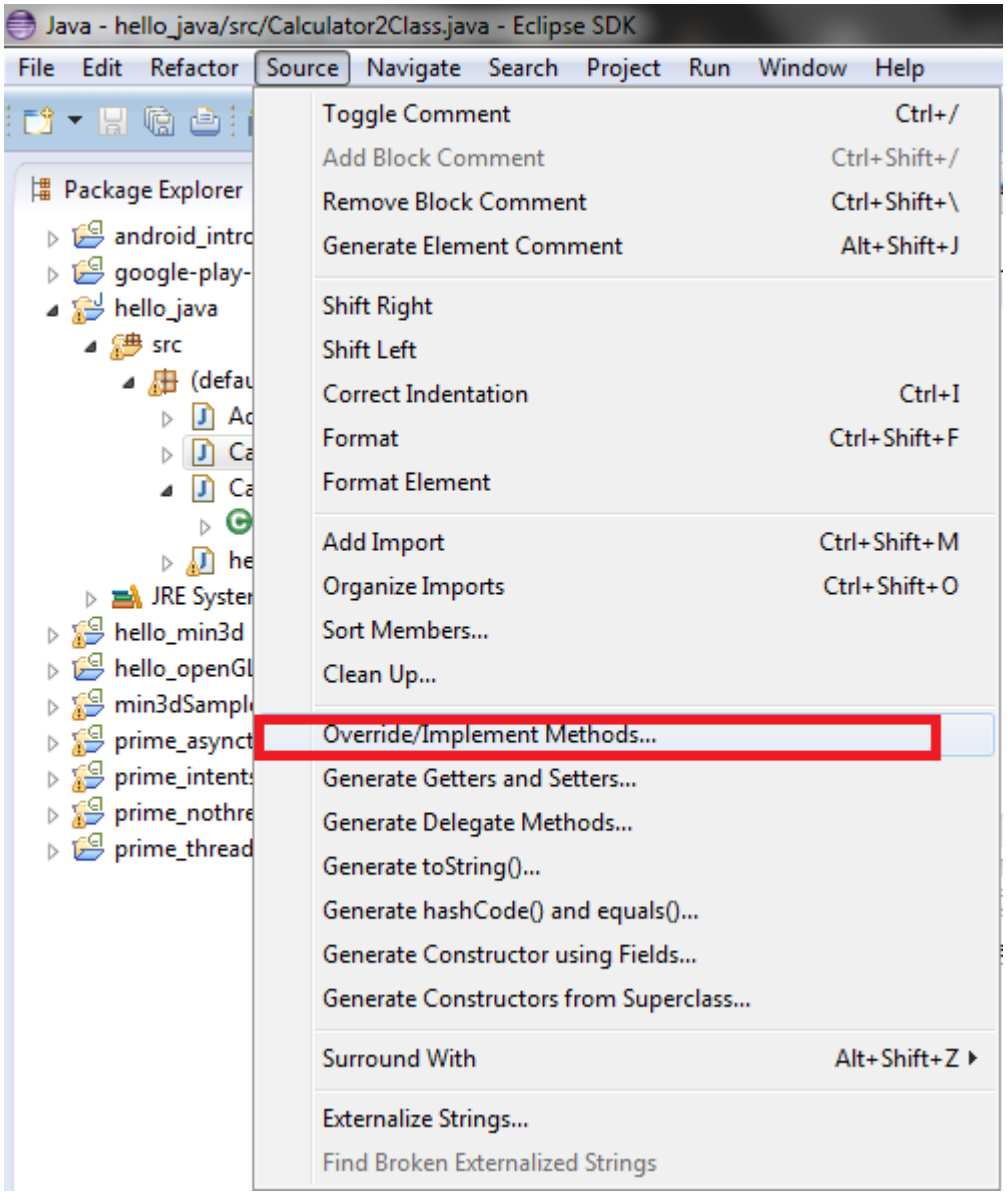
```
public int Sub(){
        return Math.abs(A-B);
}
```

Introduction to Java: Method Override

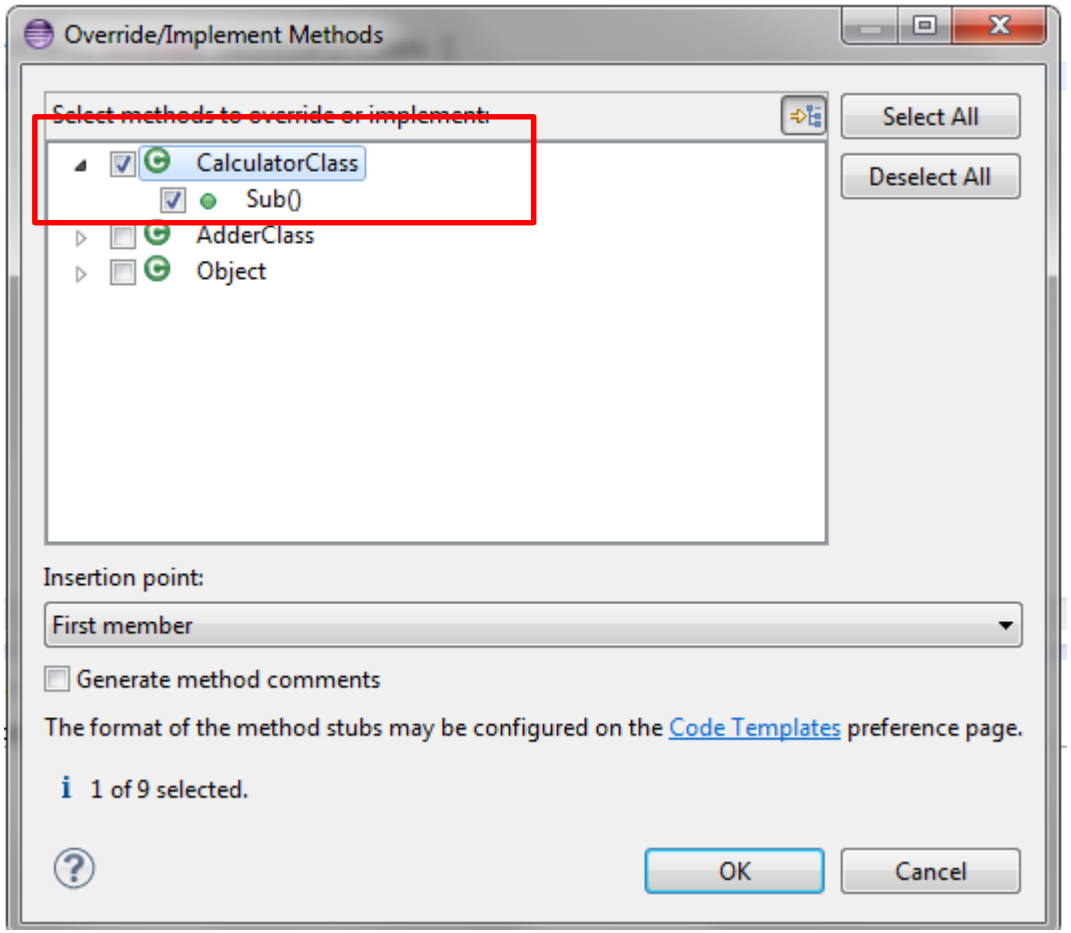



```
public int Sub(){
        return Math.abs(A-B);
}
```

Introduction to Java: Method Override



Introduction to Java: Method Override



Introduction to Java: Method Override

```
public class Calculator2Class extends CalculatorClass {

@Override
public int Sub() {
// TODO Auto-generated method stub
return super.Sub();
}

}
```

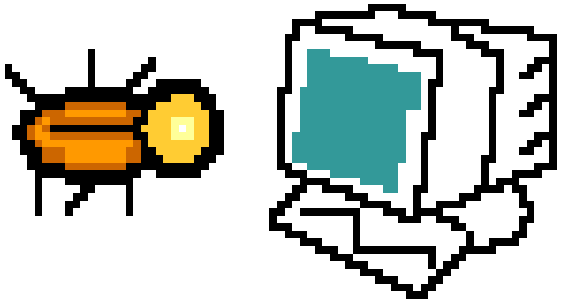
Introduction to Java: Method Override

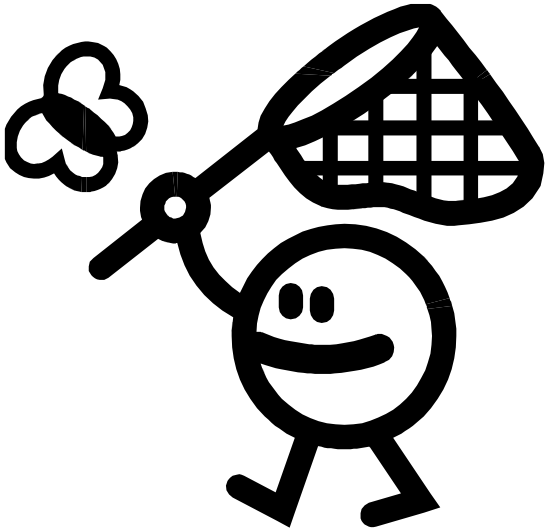
```
public class Calculator2Class extends CalculatorClass {

@Override
public int Sub() {
// TODO Auto-generated method stub
return super.Sub();
}

}
```

Introduction to Java: Method Override

```
public class Calculator2Class extends CalculatorClass {

@Override
public int Sub() {
// TODO Auto-generated method stub
return Math.abs(A-B);

}

}
```

Introduction to Java: Exception-Handling





Introduction to Java: Try - cache - finally

Format:

```
try
{
   statements that can throw exceptions
}
catch (exception-type identifier)
{
   statements executed when exception is thrown
}
finally
{
   statements that are executed whether or not
   exceptions occur
```

Introduction to Java:
Try - cache - finally

Example:

```
System.out.print("Total:");
Scanner sc=new Scanner(System.in);
int a=sc.nextInt();
System.out.print("Divided by :");
sc.nextLine();
int b=sc.nextInt();
        int c=0;
        c=a/b;
        System.out.println(c);
```

Introduction to Java:
Try - cache - finally

```
<terminated> hello_java [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (16 Jul 2013 14:50:19)
Total:10
Divided by :0
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at hello_java.main(hello_java.java:95)
```

Introduction to Java:
Try - cache - finally

Example:

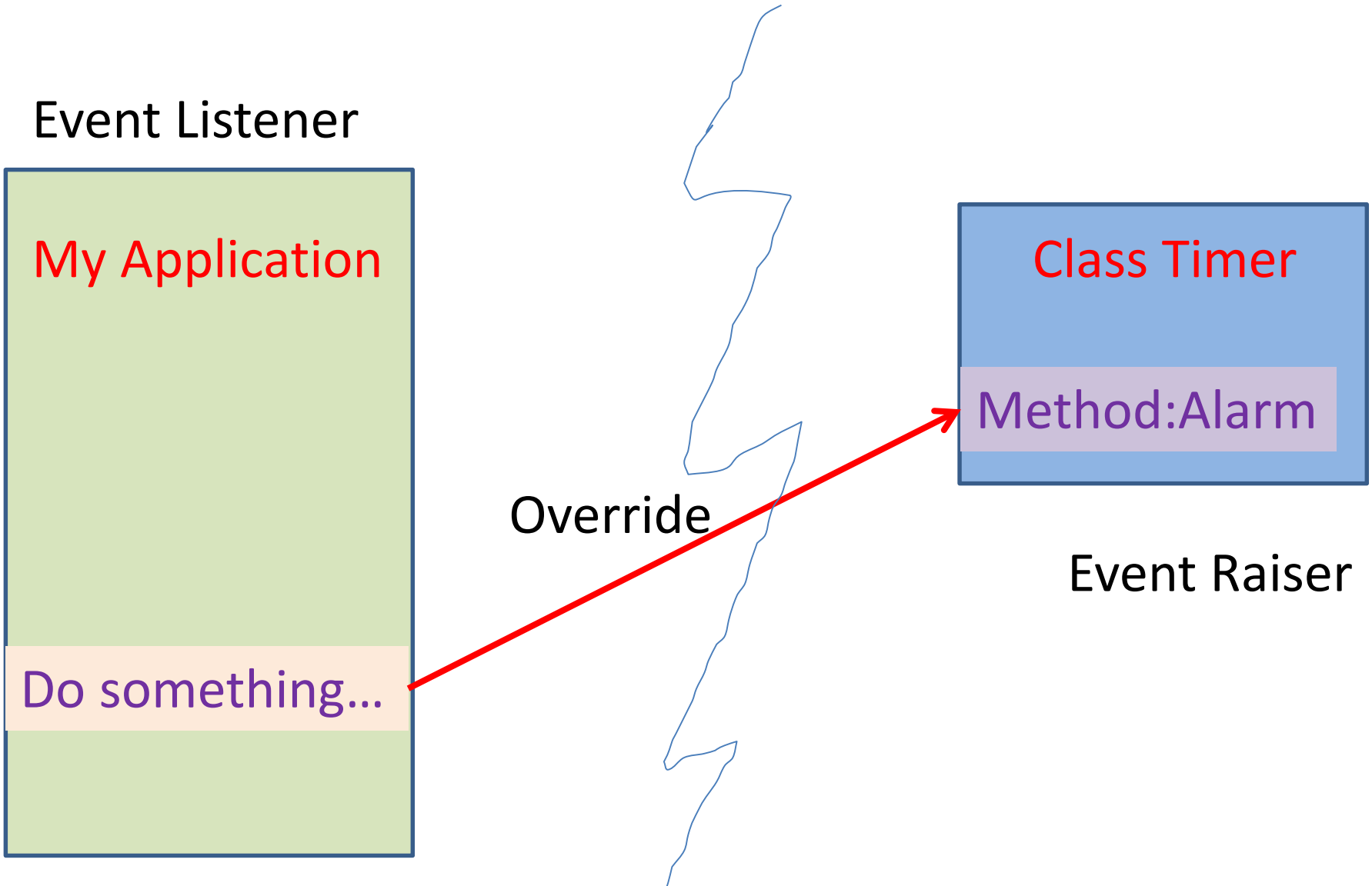
```
System.out.print("Total:");
Scanner sc=new Scanner(System.in);
int a=sc.nextInt();
System.out.print("Divided by :");
sc.nextLine();
int b=sc.nextInt();
int c=0;
    try {
            c=a/b;
        } catch (Exception e) {
            System.out.println("Bad input");
        }finally{
            System.out.println(c);
        }
```

Introduction to Java: Event Listener

Event Listener

Event Raiser

Introduction to Java: Event Listener



Introduction to Java: Event Listener

```
static TimerTask Clock = new TimerTask(){
public void run() {
 Date date = new Date();
c++;
System.out.println("Print"+ date.toString());
}
};
```

Introduction to Java: Event Listener

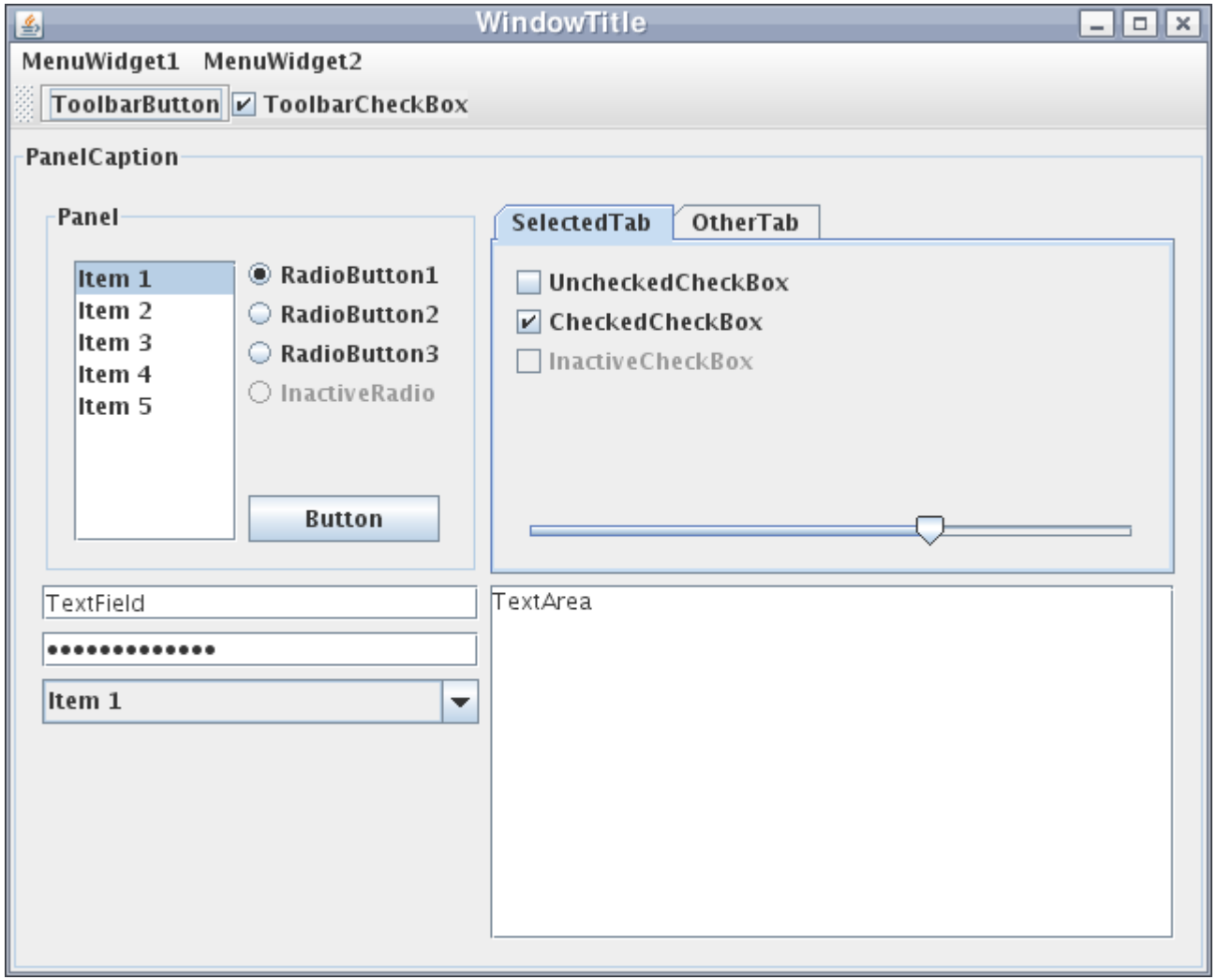
```
public class hello_timer {
static int c=0;
public static void main(String[] args) {
Timer Ticker = new Timer(true);
Ticker.scheduleAtFixedRate(Clock, 0, 1000);
for(;;)
    {
        if(c>60)
        {
        break;
        }
    }
                                        }

}
```

Swing and Windows Application

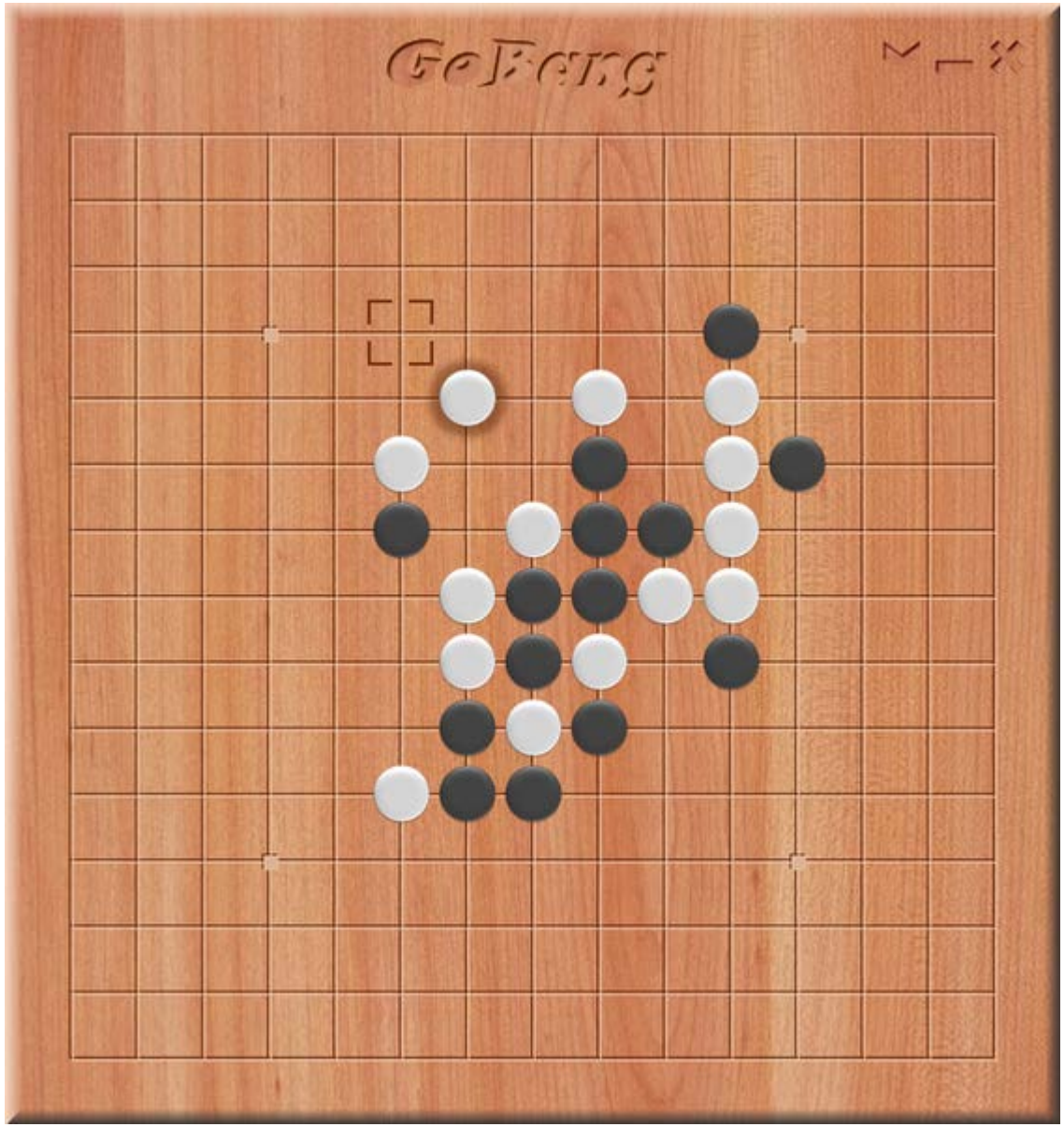
Introduction to Java

Swing is the primary Java GUI widget toolkit. It is part of Oracle's Java Foundation Classes (JFC) — an API for providing a graphical user interface (GUI) for Java programs.

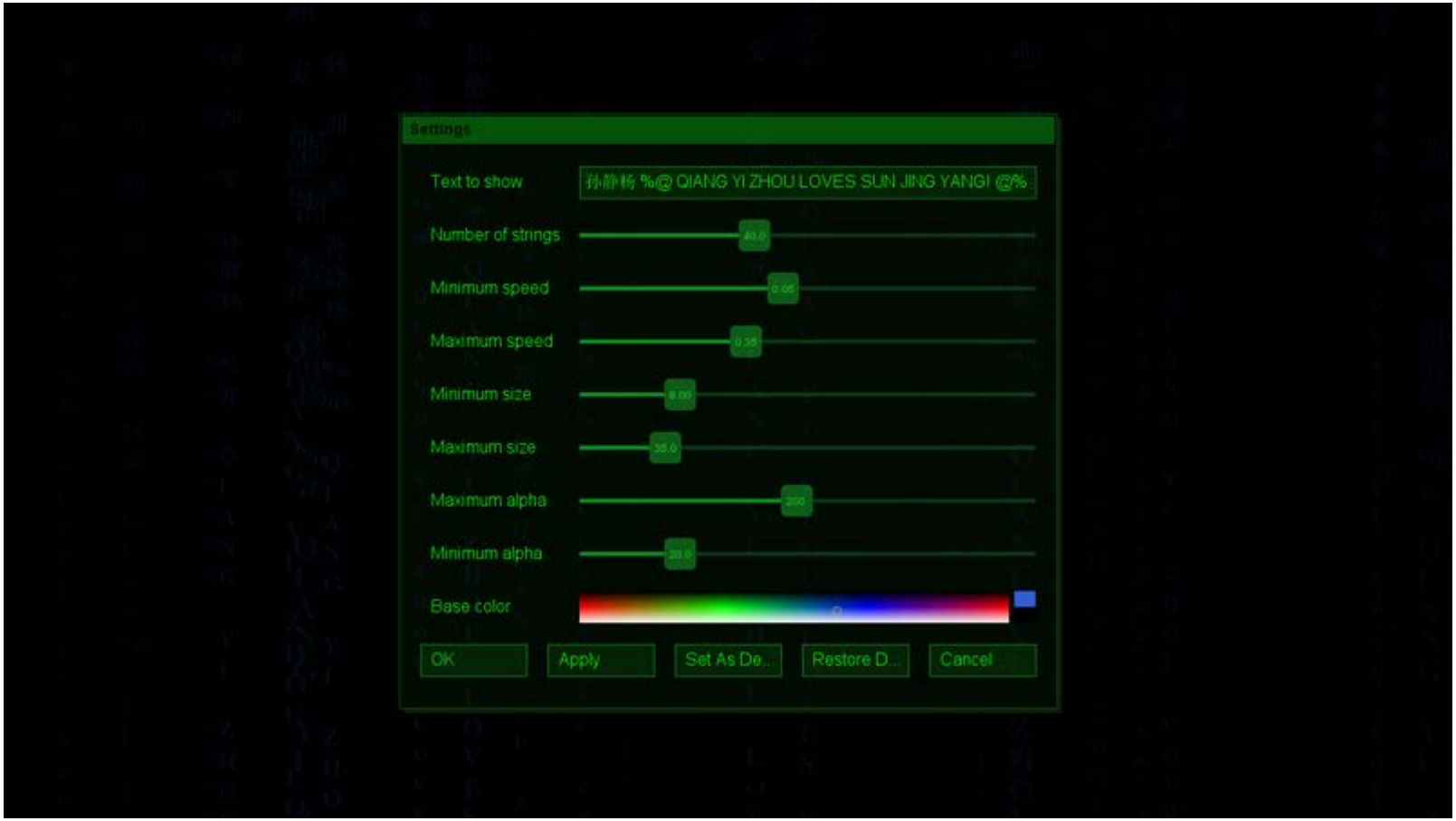
Introduction to Java



Introduction to Java

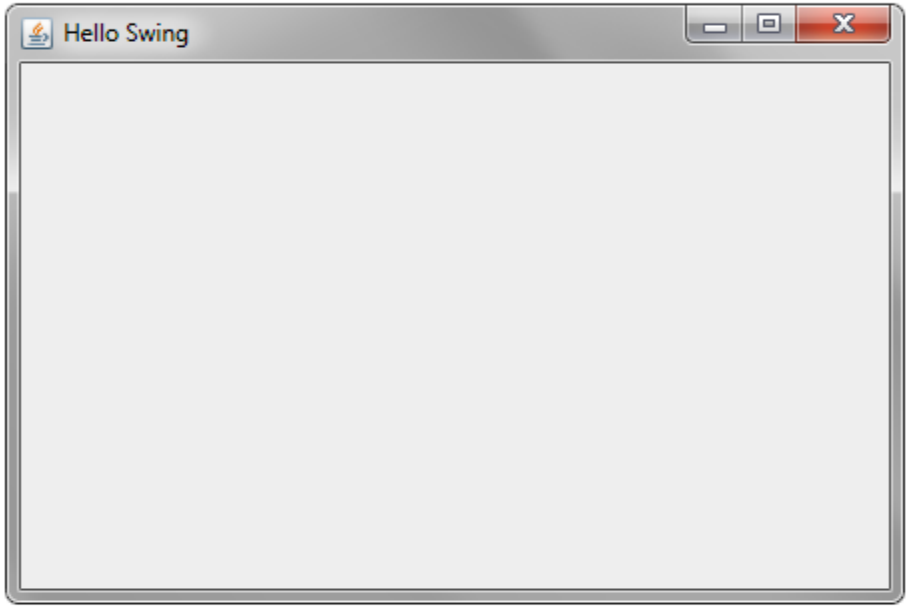


Introduction to Java



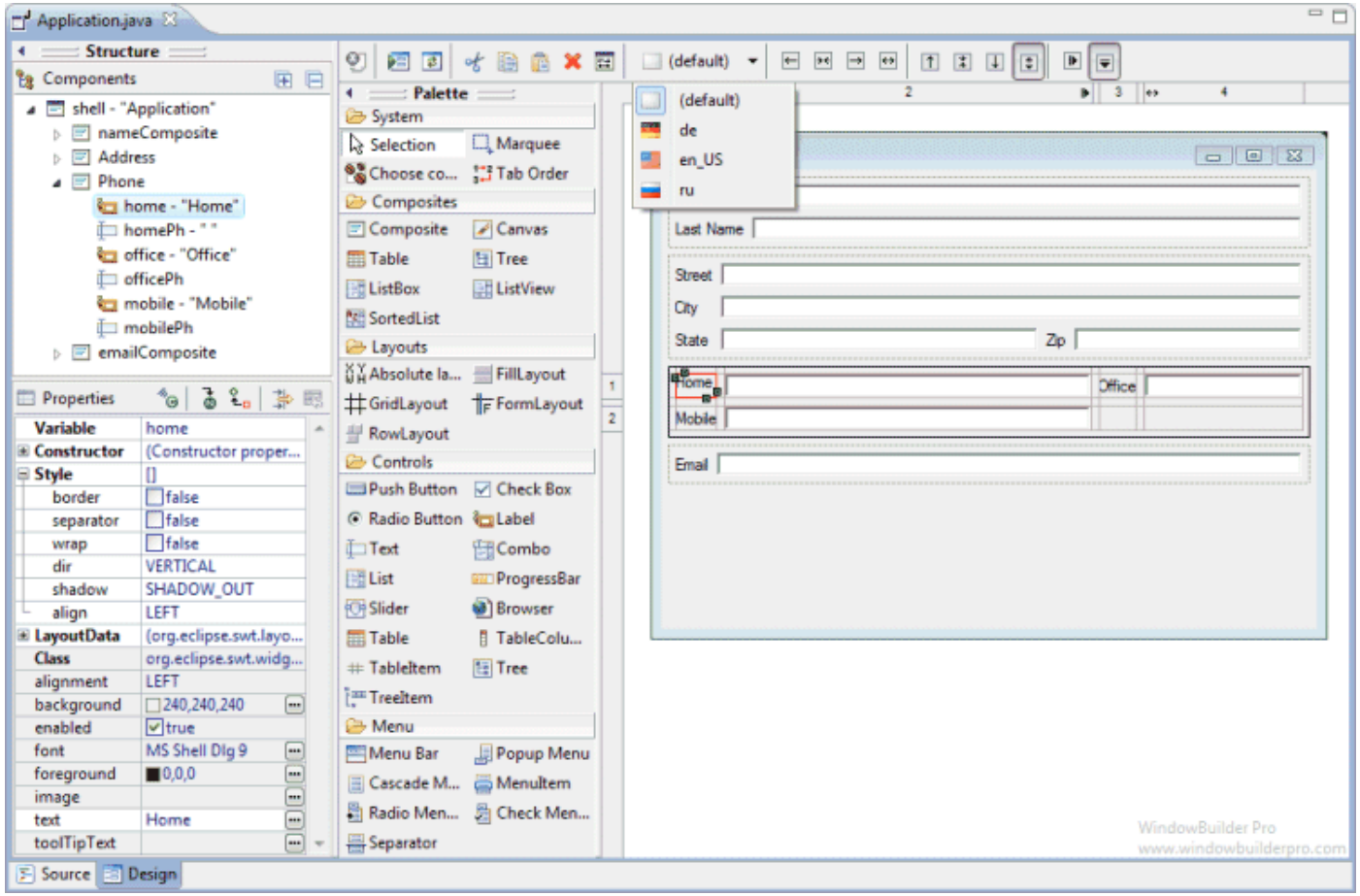
Introduction to Java

```java
import java.awt.EventQueue;


public class helloswing {

    private JFrame frame;
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    helloswing window = new helloswing();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public helloswing() {
        initialize();
    }
    private void initialize() {
        frame = new JFrame();
        frame.setBounds(100, 100, 450, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setTitle("Hello Swing");
    }

}
```



Introduction to Java

WindowBuilder - is a powerful and easy to use bi-directional Java GUI designer



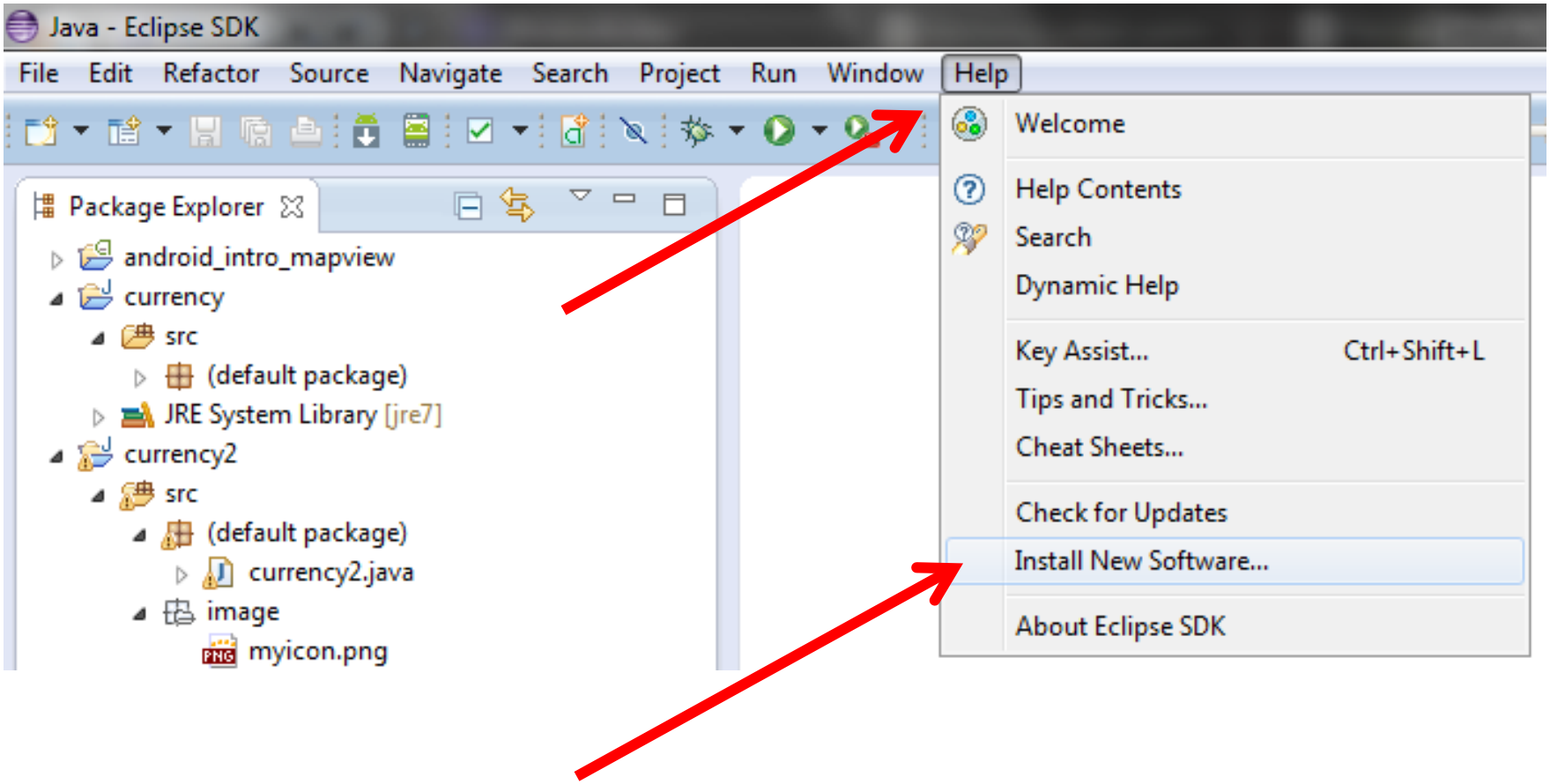
Introduction to Java

http://www.eclipse.org/windowbuilder/download.php

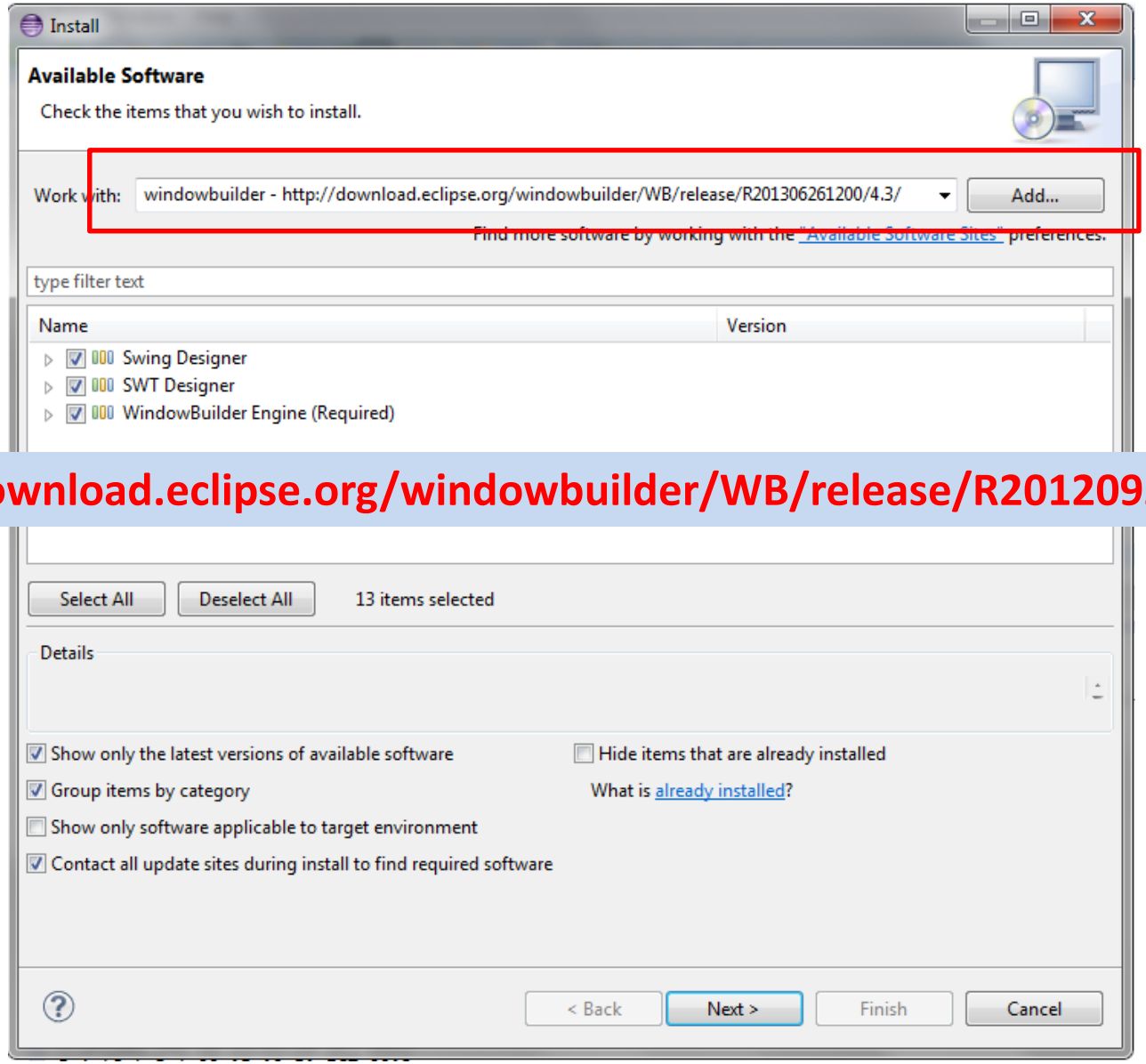
Installing WindowBuilder Pro

http://download.eclipse.org/windowbuilder/WB/release/R201209281200/4.2/

Introduction to Java



Introduction to Java

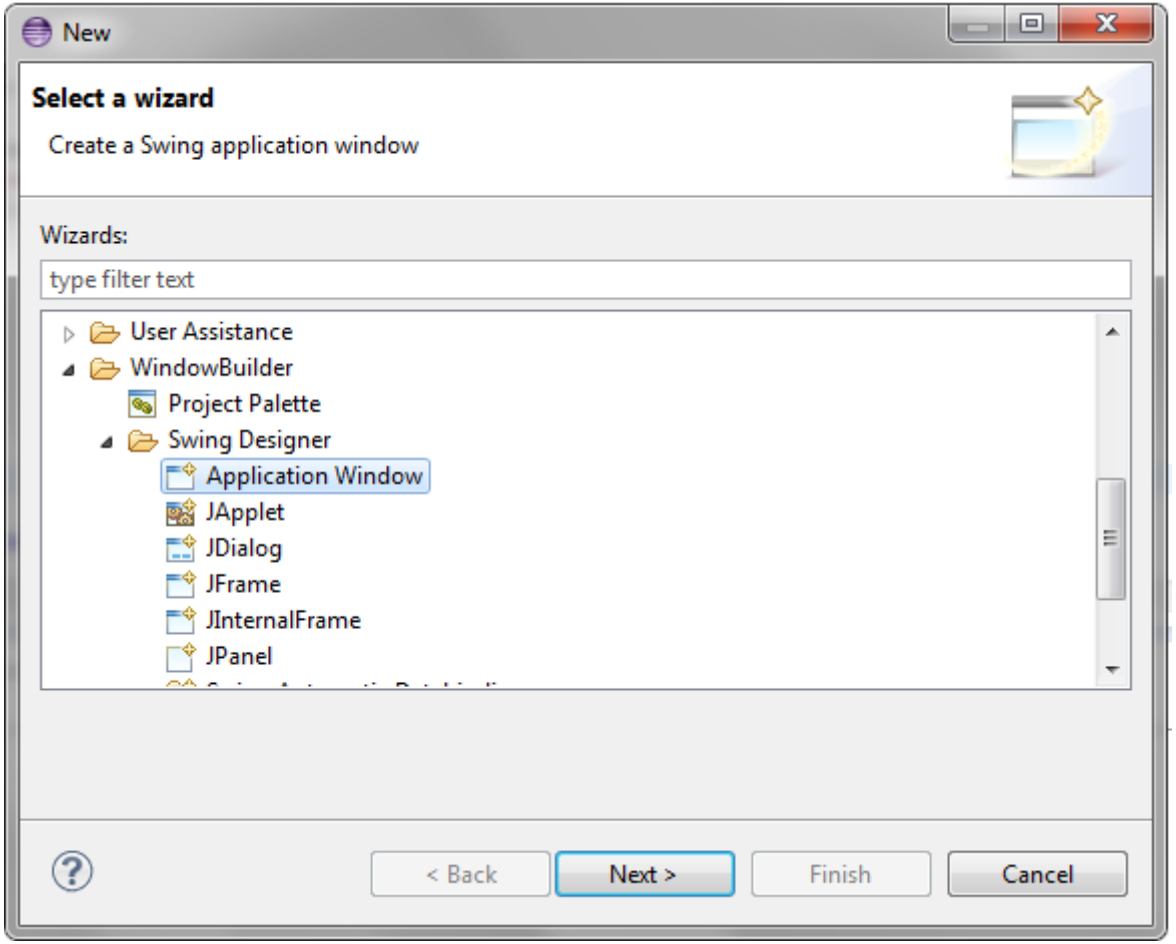


http://download.eclipse.org/windowbuilder/WB/release/R201209281200/4.2/

Introduction to Java

Building first swing application

Introduction to Java



Introduction to Java

Java - currency/src/currency.java - Eclipse SDK

File Edit Refactor Source Navigate Search Project Run Window Help

Quick Access | Java | DDMS

Packa...

- android_intro_map
- currency
  - src
    - (default pa
      - currenc
  - JRE System Lib
- google-play-servic
- hello_java
- hello_min3d
- hello_openGL
- hello_timer
- min3dSampleProje
- prime_asynctask
- prime_intentservic
- prime_nothread
- prime_thread

currency.java

```
import java.awt.EventQueue;


public class currency {

    private JFrame frame;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    currency window = new currency();
                    window.frame.setVisible(true);
                } catch (Exception e) {
```

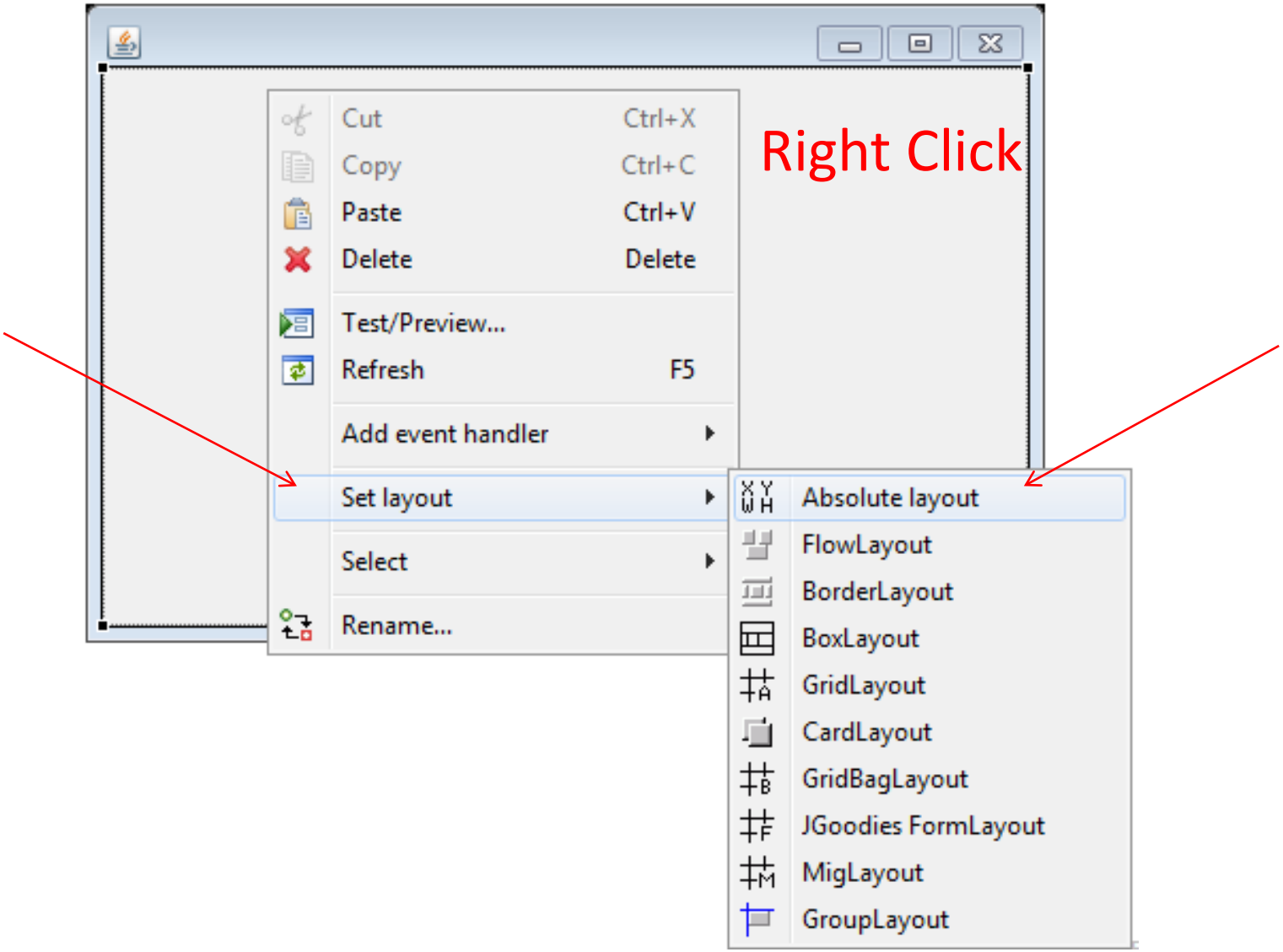
Source | Design

Problems | @ Javadoc | Declaration | Console

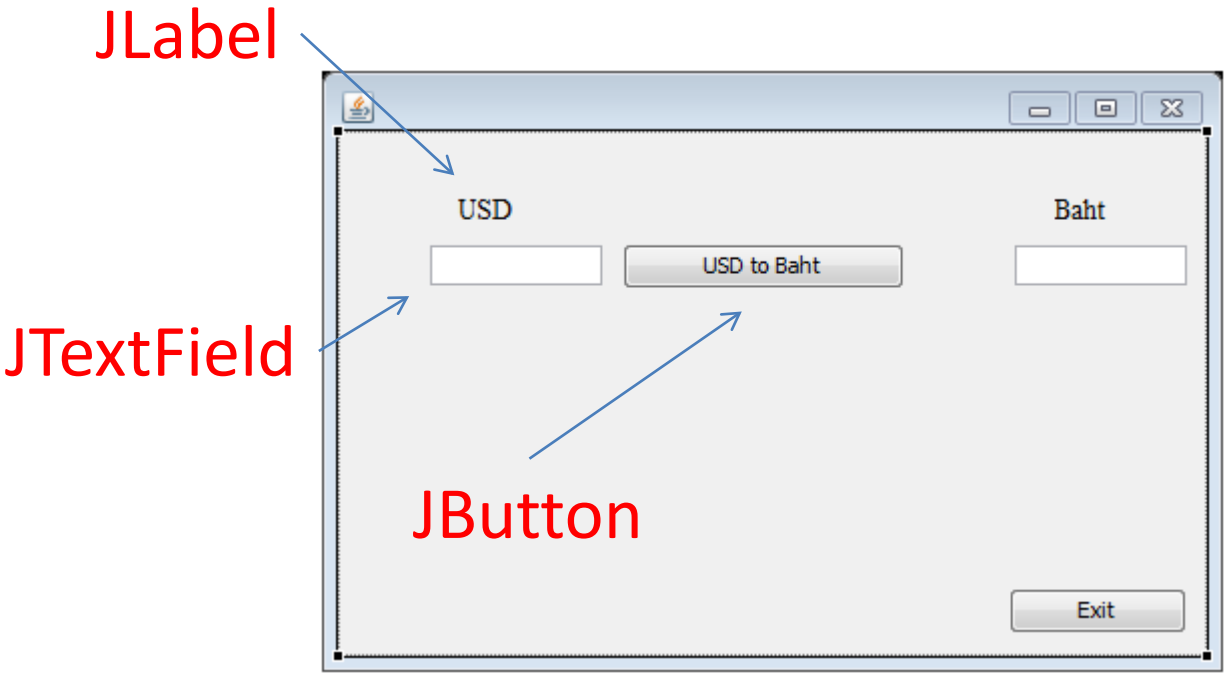
Android

Writable | Smart Insert | 17 : 1

Introduction to Java

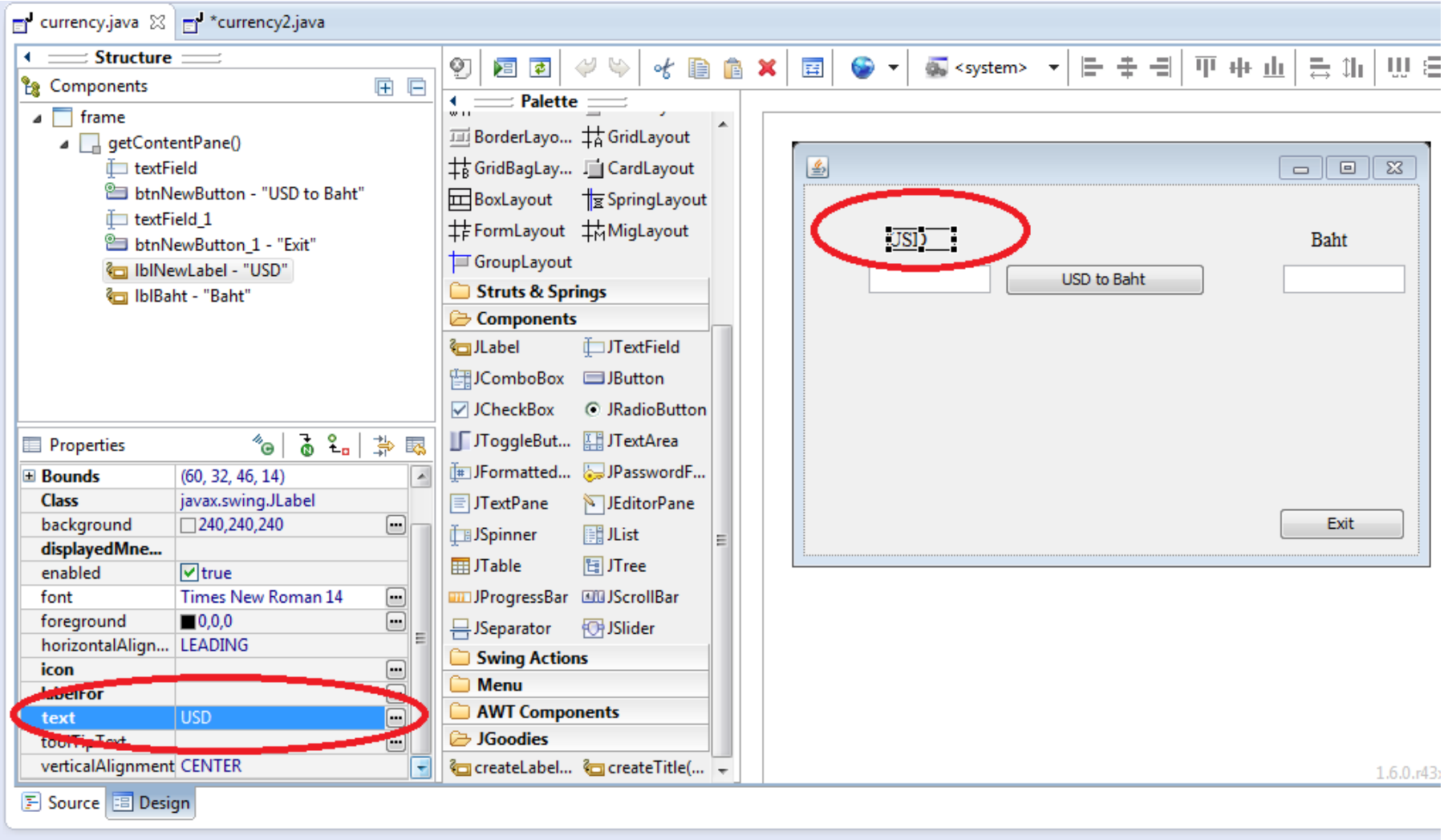


Introduction to Java



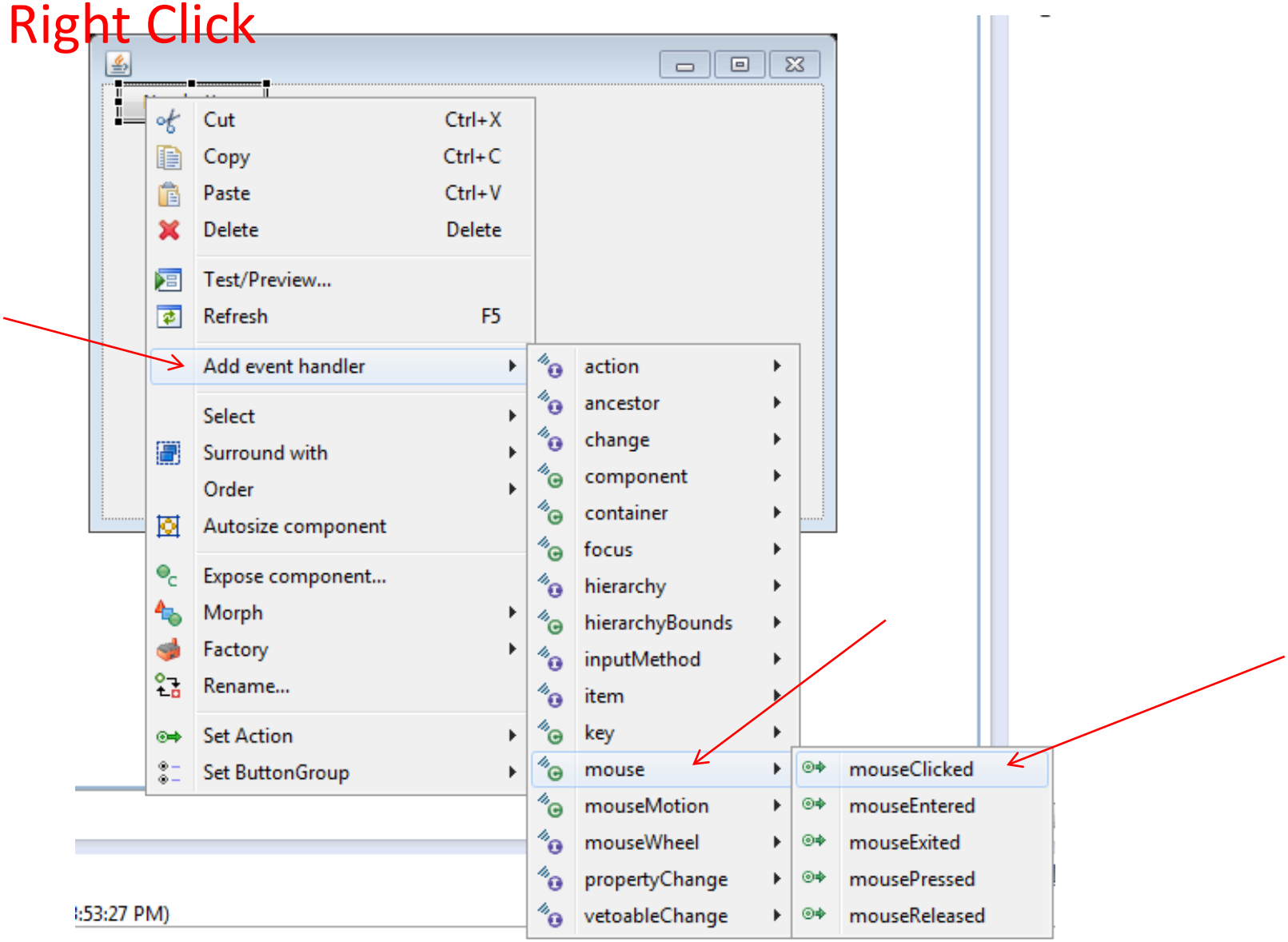
Project:Currency

Introduction to Java



Introduction to Java

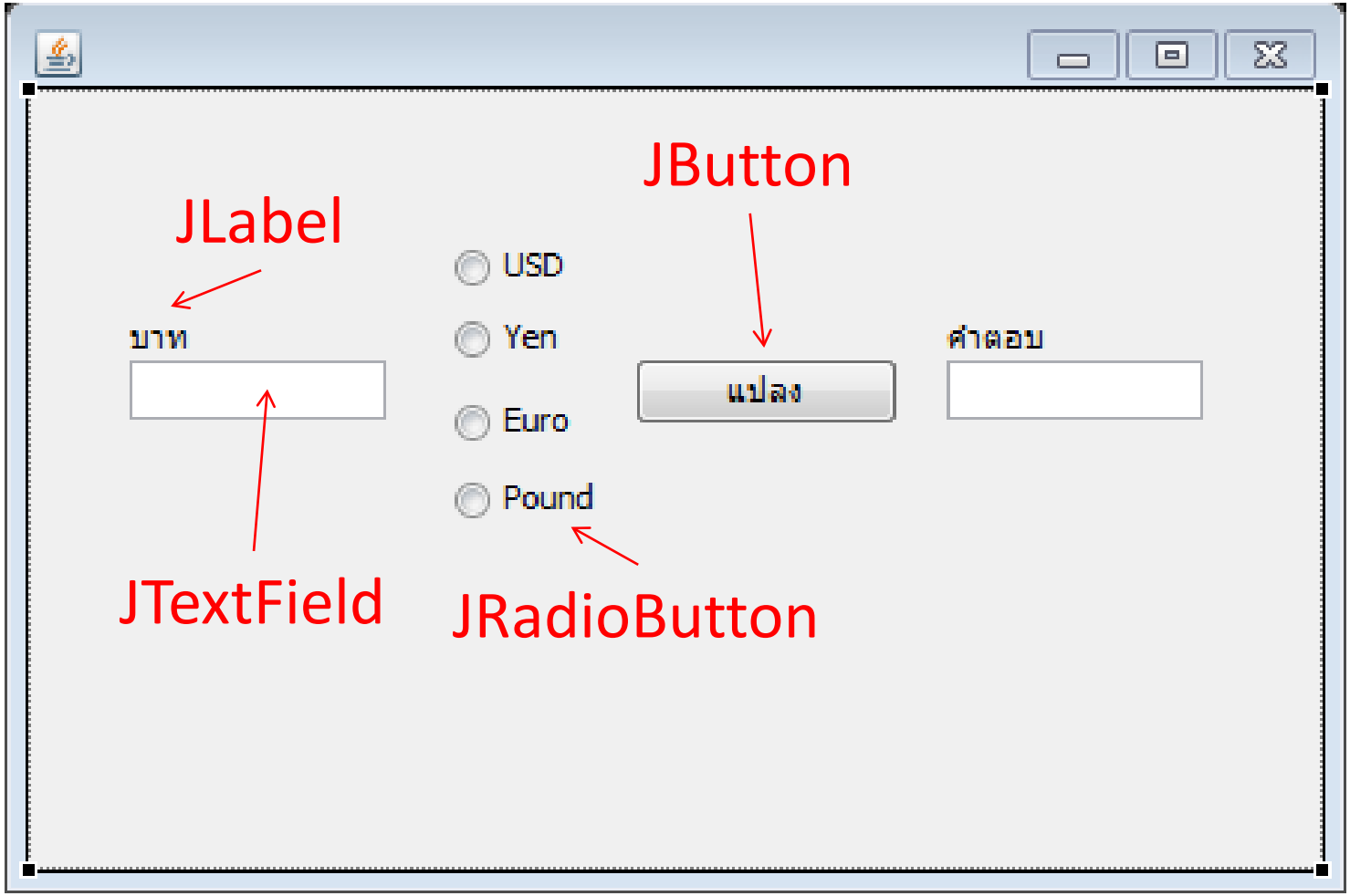
Right Click



Introduction to Java

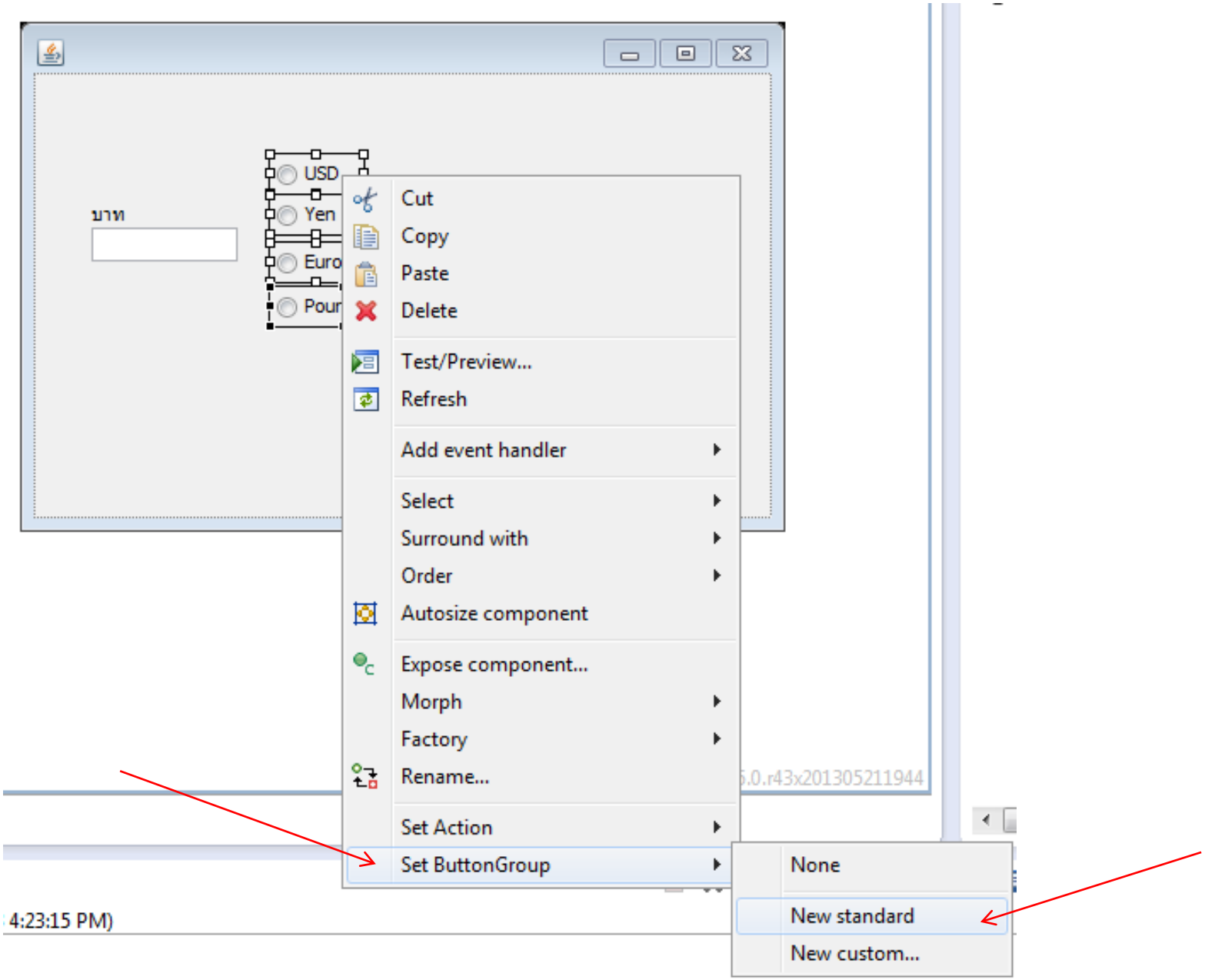
```
JButton btnNewButton = new JButton("USD to Baht");
 btnNewButton.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
        Double a=Double.valueOf(textField.getText());
        textField_1.setText(String.format("%.2f", a*30));
                        }
});
```

Introduction to Java



Project:Currency2

Introduction to Java



Introduction to Java

```
btnNewButton.addMouseListener(new
MouseAdapter() {
                    @Override
public void mouseClicked(MouseEvent e) {
Double a=Double.valueOf(textField.getText());
Double b=0d;
    if(r1.isSelected()==true)b=a*.5;
    if(r2.isSelected()==true)b=a*.6;
    if(r3.isSelected()==true)b=a*.7;
    if(r4.isSelected()==true)b=a*.8;
    textField_1.setText(String.valueOf(b));
                    }
            });
```

Introduction to Java

Project:stopwatch

Introduction to Java

```
private JFrame frame;
static JLabel lblNewLabel;
static String mode="clock";
static long starttime,stoptime;
```

Introduction to Java

```
private void initialize() {
frame = new JFrame();
frame.setBounds(100, 100, 450, 300);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.getContentPane().setLayout(null);

 lblNewLabel = new JLabel("New label");
 lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel.setForeground(Color.GREEN);
lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 26));
lblNewLabel.setBounds(31, 45, 354, 94);
frame.getContentPane().add(lblNewLabel);
```

Introduction to Java

```
JButton btnNewButton = new JButton("START");
btnNewButton.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
starttime=System.currentTimeMillis();
mode="tick";

}
});
btnNewButton.setBounds(0, 185, 89, 23);
frame.getContentPane().add(btnNewButton);

JButton btnNewButton_1 = new JButton("STOP");
btnNewButton_1.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
stoptime=System.currentTimeMillis();
mode="stop";
}
});
btnNewButton_1.setBounds(140, 185, 89, 23);
frame.getContentPane().add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("RESET");
btnNewButton_2.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
mode="clock";
}
```

Introduction to Java

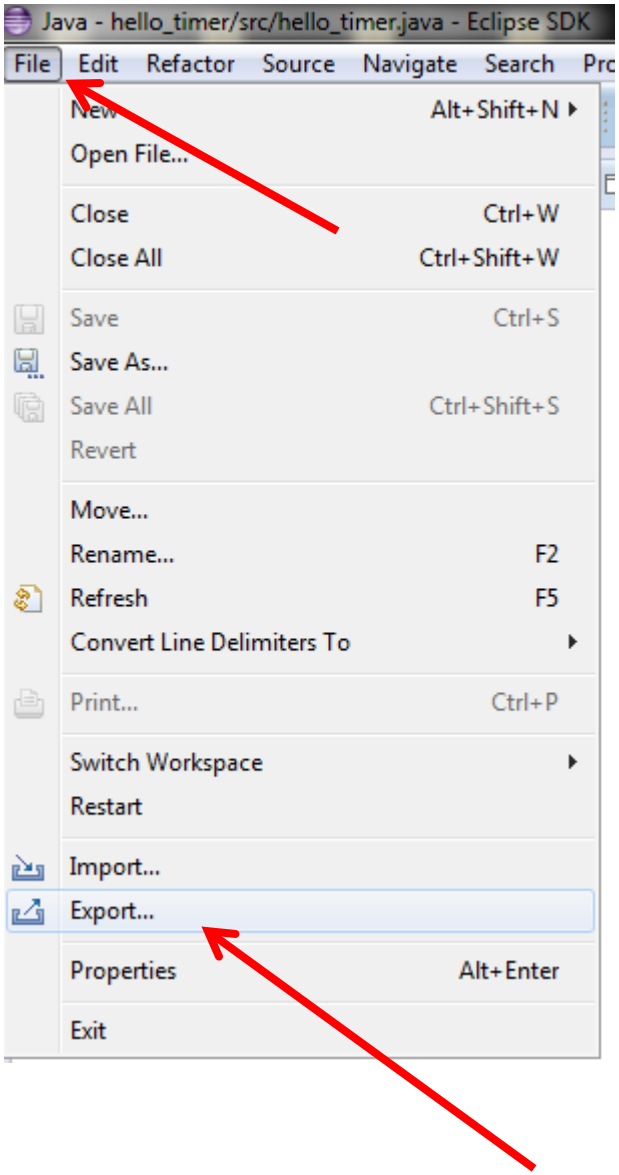
```
Timer Ticker = new Timer(true);
    Ticker.scheduleAtFixedRate(Clock, 0, 10);
```

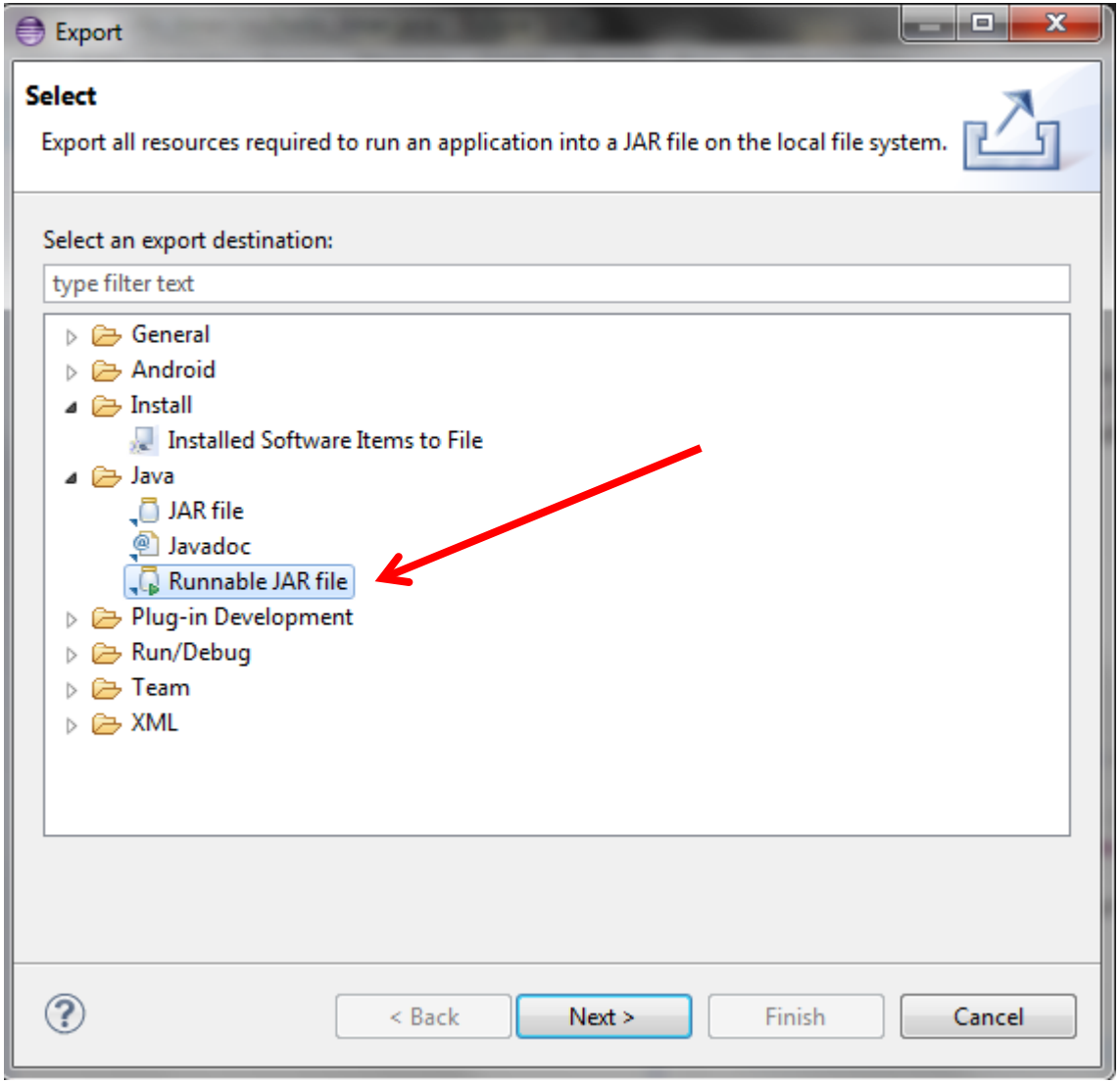
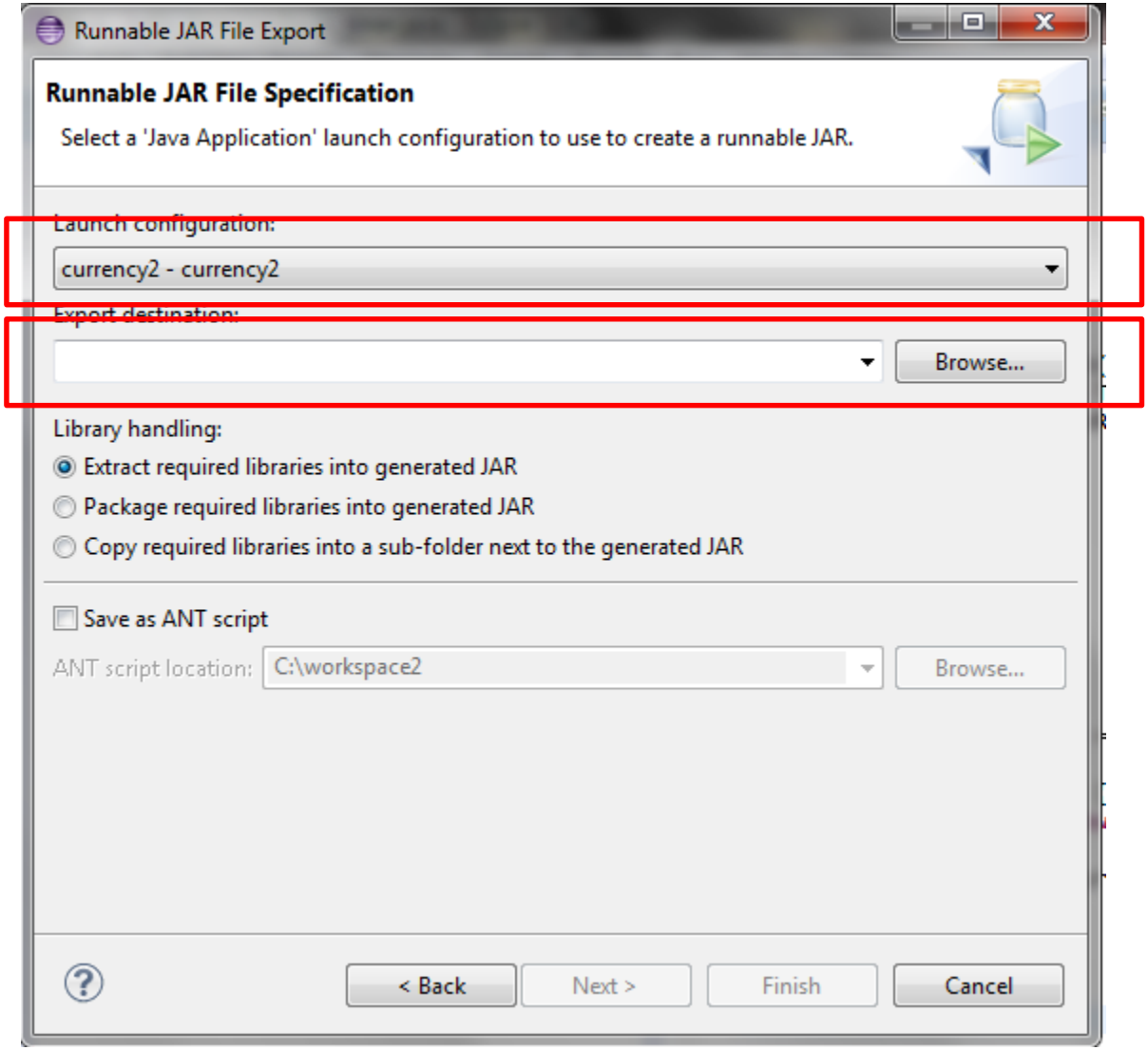
Introduction to Java

```
static TimerTask Clock = new TimerTask(){

 public void run() {
 Date date = new Date();

 SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss.SSS");
 if(mode.equals("clock")) lblNewLabel.setText(sdf.format(date));

 if(mode.equals("tick"))
 {
date=new Date(System.currentTimeMillis()-starttime);
sdf = new SimpleDateFormat("mm:ss.SSS");
 lblNewLabel.setText(sdf.format(date));

 }

 if(mode.equals("stop"))
 {
date=new Date(stoptime-starttime);
    sdf = new SimpleDateFormat("mm:ss.SSS");
 lblNewLabel.setText(sdf.format(date));

 }
 }
};
```

Export executable program

Introduction to Java





Introduction to Java