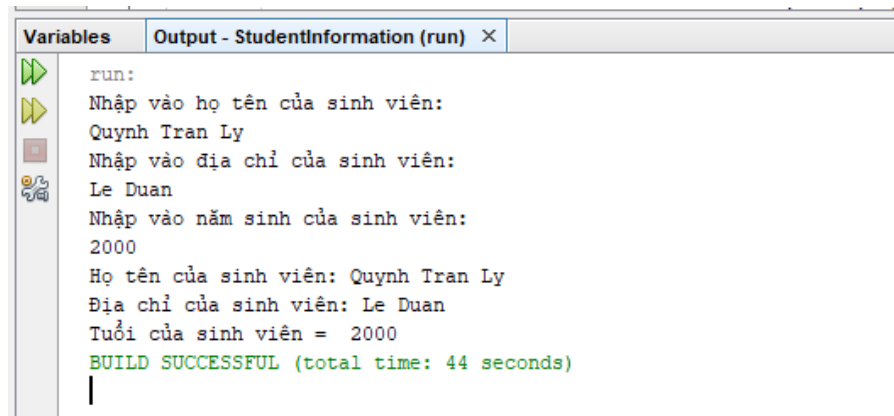


# Bài tập tại lớp 1

-----  -----  
**Tên Sinh Viên** \_\_\_\_\_ **Lớp** \_\_\_\_\_ **Mã SV** \_\_\_\_\_  
**Ngày** \_\_\_\_\_ **Điểm** \_\_\_\_\_

BÀI 1: Viết chương trình cho phép nhập thông tin sinh viên từ bàn phím sau đó xuất ra màn hình với định dạng như sau



```
run:
Nhập vào họ tên của sinh viên:
Quynh Tran Ly
Nhập vào địa chỉ của sinh viên:
Le Duan
Nhập vào năm sinh của sinh viên:
2000
Họ tên của sinh viên: Quynh Tran Ly
Địa chỉ của sinh viên: Le Duan
Tuổi của sinh viên = 2000
BUILD SUCCESSFUL (total time: 44 seconds)
```

## HƯỚNG DẪN:

- ✓ Tạo lớp chứa phương thức `public static void main(String[] args){...}` `public class StudentInformation{`  
    `public static void main(String[] args){`  
        `}`  
    `}`
- ✓ Viết mã cho `main()`: Sử dụng `Scanner` để đọc dữ liệu từ bàn phím `Scanner scanner = new Scanner(System.in); System.out.print("Họ và tên: ");`  
    `name = scanner.nextLine();`
- ✓ Sử dụng `System.out.print()`, `System.out.printf()` và `System.out.println()` để xuất ra màn hình

## Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp

```
8 import java.util.Scanner;
9
10 /**
11 *
12 * @author Quynh Tran Ly
13 */
```

```

14 public class StudentInformation {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         String _____
22         int _____
23         String _____
24
25         Scanner scanner = new _____
26         System. _____
27         name = _____
28         System.out.println _____
29         address = _____
30         System.out.println _____
31         _____ = scanner.nextInt();
32
33
34         System.out.println("Họ tên _____
35         System.out.println("Địa chỉ _____
36         System.out.println("Tuổi _____
37     }
38
39 }
40

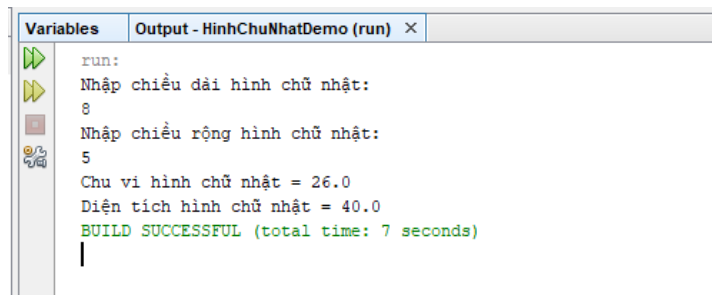
```

## BÀI 2 (2 ĐIỂM)

Viết chương trình nhập từ bàn phím 2 cạnh của hình chữ nhật. Tính và xuất chu vi, diện tích và cạnh nhỏ của hình chữ nhật.

### HƯỚNG DẪN:

- ✓ Chu vi = (dài + rộng)\*2
- ✓ Diện tích = dài \* rộng



**Sinh viên hoàn thiện phần Code sau dưới sự hướng dẫn của giảng viên tại lớp**

```

8 import java.util.Scanner;
9 /**
10  *
11  * @author Quynh Tran Ly
12  */
13 public class _____
14
15  /**
16   * @param args the command line arguments
17   */
18  public static void main(String[] args) {
19      // TODO code application logic here
20      double chieuDai _____
21      Scanner _____
22      _____
23      _____
24
25      System.out.println("Nhập chiều rộng hình chữ nhật: ");
26      _____
27      chuVi = _____
28      dienTich = _____
29      System.out.println("Chu vi hình chữ nhật = ");
30      System.out.println("Diện tích hình chữ nhật = ");
31  }
32
33 }
34

```

**Các phím tắt trong NetBeans**

Tìm kiếm và thay thế

Ctrl-F3: tìm từ tại vị trí con trỏ chuột

F3/Shift-F3: tìm từ kế tiếp/trước đó trong file

Ctrl-F/H: tìm/thay thế từ trong file

Alt-F7: tìm vị trí hàm được sử dụng trong projects

Ctrl-R: đổi tên biến tại con trỏ chuột

Alt-Shift-L: copy đường dẫn của file đang mở

Coding trong Java

Alt-Insert: sinh ra code

Ctrl-Shift-I: tự động imports class cần thiết

Alt-Shift-F: định dạng code

Ctrl-Shift-Up/D: copy dòng tại con trỏ chuột

Ctrl-/: bật tắt comment

Ctrl-E/X: xóa dòng tại con trỏ chuột

Biên dịch, kiểm tra và chạy project

F9: biên dịch package/file

F11: build project chính

Shift-F11: xóa và build project chính

Shift-F6: chạy file hiện tại

F6: chạy project

Debugging

Ctrl-F5: bắt đầu debugging project chính

Ctrl-Shift-F5: bắt đầu debugging file hiện tại

Ctrl-Shift-F6: bắt đầu debugging file

Ctrl-Shift-F8: đặt breakpoint mới

Khác

Tool-->Options--> KeyMaps: chọn Generate Javadoc: tự động tạo javadoc của project cho bạn

Một số phím tắt thú vị khác

## **CÁC PHÍM TẮT GỖ LỆNH – Java Editor Code Templates**

LƯU Ý nhớ ấn thêm phím tab và có phân biệt chữ hoa vd: En + tab =Enumeration

En Enumeration

Ex Exception

Ob Object

Psf public static final

Psfb public static final boolean

Psfi public static final int

Psfs	public static final String
St	String
ab	abstract
as	assert true;
bcom	/**/
bo	boolean
br	break;
ca	catch (
cl	class
cn	continue
db	double
df	default:
dowhile	do { } while (condition);
eq	equals
ex	extends
fa	false
fcom	// <editor-fold></editor-fold>
fi	final
fl	float
forc	for (Iterator it = collection.iterator();it.hasNext()); { Object elem = (Object) it.next();}
fore	for (Object elem : iterable) { }
fori	for (int i = 0; i < arr.length; i++) { }
forl	for (int i = 0; i < lst.size(); i++) { }
Object	object = lst.get(i); }
forst	for (StringTokenizer st = new
String	Tokenizer(""); st.hasMoreTokens();) { }
forv	for (int i = 0; i < vct.size(); i++) { }
Object	object = vct.elementAt(i);}
fy	finally{  }
ie	interface
ifelse	if (condition){ }else { }
iff	if (exp) { }
im	implements
inst	if (exp instanceof Object) { }
Object	obj = (Object) exp;
iof	instanceof
ir	import
le	length
na	native
newo	Object name = new Object(args);
pe	protected
pr	private
psf	private static final
psfb	private static final boolean
psfi	private static final int
psfs	private static final String

pst	printStackTrace();
psvm	public static void main(String[] args){}
pu	public
re	return
runn	Runnable runnable = new Runnable() {
public	void run() {}};
serr	System.err.println(" ");
sh	short
sout	System.out.println(" ");
soutv	System.out.println("Object = " + Object);
st	static
su	super
sw	switch (var) { case val: break;
default:	throw new AssertionError();}
sy	synchronized
tds	Thread.dumpStack();
th	throws
tr	transient
trycatch	try {}
catch	(Exception e) {}
tw	throw
vo	volatile
wh	while (
whileit	while (it.hasNext()) {
Object	elem = (Object) it.next();}
whilen	while (en.hasMoreElements()) {
Object	elem = (Object) en.nextElement();}
whilexp	while (exp) {}
ag	application.getAttribute(" ")
ap	application.putAttribute(" ",)
ar	application.removeAttribute(" ")
cfgi	config.getInitParameter(" ")
jspf	<jsp:forward page=" "/>
jspg	<jsp:getProperty name=" " property=""/>
jspi	<jsp:include page=" "/>
jsp	<jsp:plugin type=" " code=""/></jsp:plugin>
jsp	<jsp:setProperty name=" " property=""/>
jspu	<jsp:useBean id=" " type=""/>
oup	out.print(" ")
oupl	out.println(" ")
pcg	pageContext.getAttribute(" ")
pcgn	pageContext.getAttributeNamesInScope(" ")
pcgs	pageContext.getAttributeScope(" ")
pcr	pageContext.removeAttribute(" ")
pc	s pageContext.setAttribute(" ",)
pg	<% @page  %>

pga	<%@page autoFlush="false"%>
pgb	<%@page buffer=" kb"%>
pgc	<%@page contentType=" "%">
pgerr	<%@page errorPage=" "%">
pgex	<%@page extends=" "%">
pgie	<%@page isErrorPage="true"%>
pgim	<%@page import=" "%">
pgin	<%@page info=" "%">
pgit	<%@page isThreadSafe="false"%>
pgl	<%@page language="java"%>
pgs	<%@page session="false"%>
rg	request.getParameter(" ")
sg	session.getAttribute(" ")
sp	session.setAttribute(" ", )
sr	session.removeAttribute(" ")
tglb	<%@taglib uri=" "%">