

BÀI GIẢNG CƠ SỞ LẬP TRÌNH

CHƯƠNG 2.

BIẾN VÀ BIỂU THỨC

NGUYỄN THÀNH THỦY

BỘ MÔN TIN HỌC QUẢN LÝ

TRƯỜNG ĐẠI HỌC KINH TẾ, ĐẠI HỌC ĐÀ NẴNG

THUYNT@DUE.EDU.VN

NỘI DUNG

- ❑ Biểu thức
- ❑ Giá trị và kiểu dữ liệu
- ❑ Biến
- ❑ Định danh
- ❑ Các hàm nhập, xuất dữ liệu cơ bản

BIỂU THỨC

❑ Toán tử toán học

- Độ ưu tiên từ cao đến thấp

Toán tử	Ý nghĩa	Ví dụ	Kết quả
**	Bình phương	$2 ** 3$	8
%	Chia lấy dư	$5 \% 2$	1
//	Chia lấy nguyên	$5 // 2$	2
/	Chia	$5 / 2$	2.5
*	Nhân	$5 * 3$	15
-	Trừ	$5 - 2$	3
+	Cộng	$2 + 2$	4

BIỂU THỨC

□ Độ ưu tiên toán tử toán học trong biểu thức

- Là trình tự xử lý của máy tính khi có đồng thời nhiều toán tử trong một biểu thức.

$(5 - 1) * ((7 + 1) / (3 - 1))$



$4 * ((7 + 1) / (3 - 1))$



$4 * (8) / (3 - 1)$



$4 * (8) / (2)$



$4 * 4.0$



16.0

GIÁ TRỊ VÀ KIỂU DỮ LIỆU

- ❑ **Giá trị (value):** là những dữ liệu cơ bản mà máy tính có thể tiếp nhận để xử lý như: **một số, một chuỗi ký tự,...**;
- ❑ **Kiểu dữ liệu (type):** là kiểu của dữ liệu được đưa vào máy tính, hoặc kiểu (*phương pháp*) tổ chức các tập dữ liệu trên máy tính;

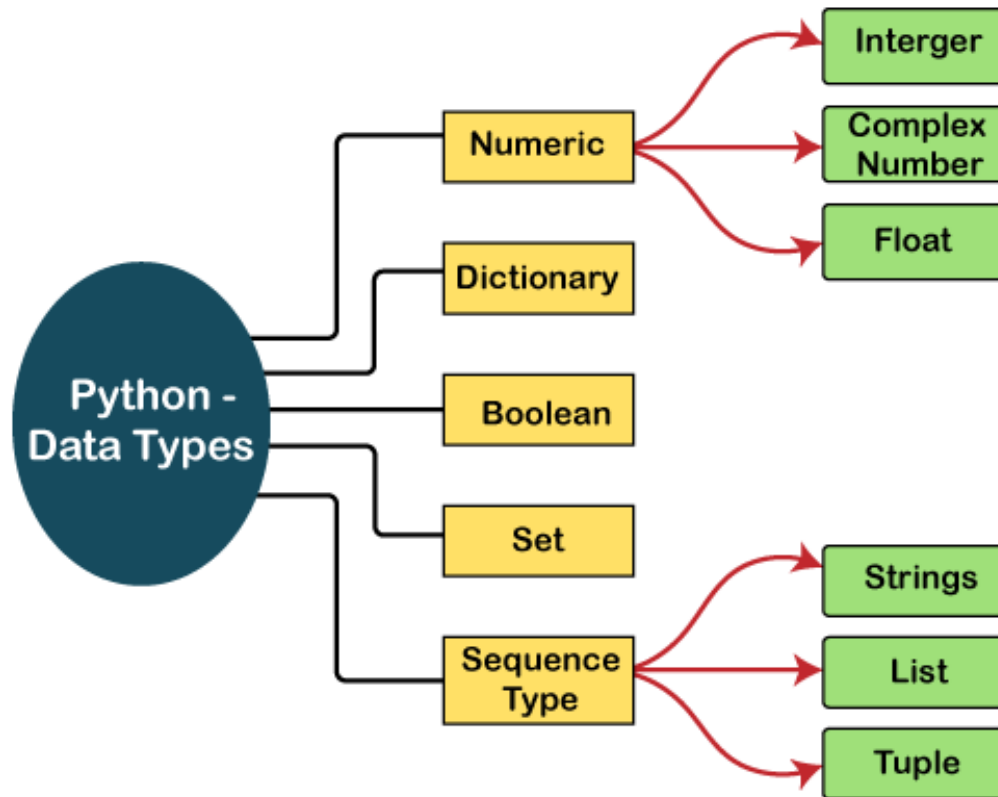
Data type	Examples
Integers	-2, -1, 0, 1, 2, 3, 4, 5
Floating-point numbers	-1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25
Strings	'a', 'aa', 'aaa', 'Hello!', '11 cats'

Nguồn: Automate the Boring Stuff with Python

- Chuỗi được đặt trong cặp dấu “ ” hoặc ‘ ’

GIÁ TRỊ VÀ KIỂU DỮ LIỆU

❑ Các kiểu dữ liệu trong Python



Hình: nguồn javatpoint.com

GIÁ TRỊ VÀ KIỂU DỮ LIỆU

❑ Kiểu dữ liệu (type):

- Chuỗi được đặt trong cặp dấu “ ” hoặc ‘ ’
- Phép nối chuỗi: + (dấu cộng)
- Phép nhân bản chuỗi: * (dấu nhân)

```
>>> 'Co so' + 'lap trinh'
```

```
'Co solap trinh'
```

```
>>> 'Python' * 5
```

```
'PythonPythonPythonPythonPython'
```

```
>>> 'Co so' + 123
```

Sai: chuỗi không cộng được với số

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: must be str, not int
```

```
>>> 'Python' * 5.0
```

Sai: phải là số nguyên

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: can't multiply sequence by non-int of type 'float'
```

```
>>> 'Lap trinh' * 'Python'
```

Sai: phải là số nguyên

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: can't multiply sequence by non-int of type 'str'
```

BIẾN (VARIABLE)

□ Khái niệm về biến:

- Biến là một đại lượng mang giá trị và có thể thay đổi trong quá trình xử lý của chương trình;
- Một biến được sử dụng để lưu trữ dữ liệu trong một chương trình, và được khai báo với một kiểu dữ liệu nhất định;
- Ví dụ:
 - **Tuoi** (tuổi) của một sinh viên
 - **DiaChi** (địa chỉ) của một khách hàng
 - **TienLuong** (tiền lương) của một nhân viên, ...

BIẾN (VARIABLE)

❑ Khái niệm về biến:

- Một biến sẽ có **một tên** và có thể chứa **một giá trị**;
- Phép gán: `=` (dấu bằng)

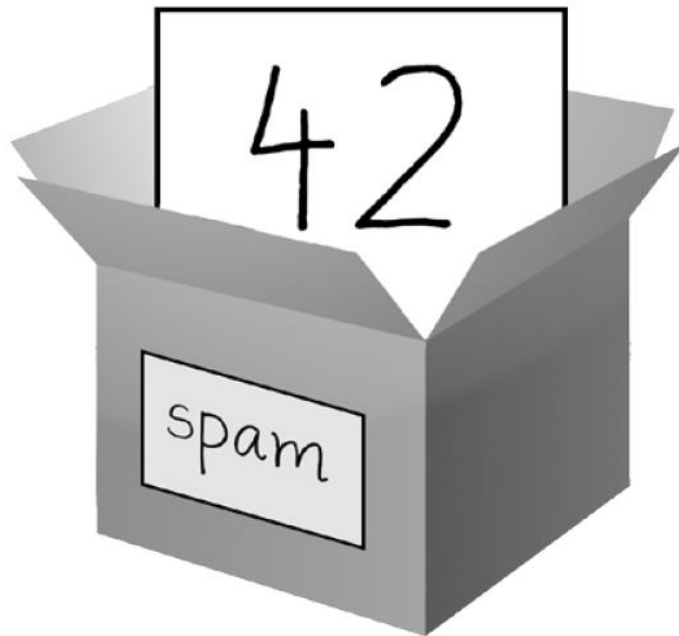


Figure 1-2: `spam = 42` is like telling the program, "The variable `spam` now has the integer value 42 in it."

BIẾN (VARIABLE)

❑ Khái niệm về biến:

For example, enter the following into the interactive shell:

```
❶ >>> spam = 40
>>> spam
40
>>> eggs = 2
❷ >>> spam + eggs
42
>>> spam + eggs + spam
82
❸ >>> spam = spam + 2
>>> spam
42
```

A variable is *initialized* (or created) the first time a value is stored in it ❶. After that, you can use it in expressions with other variables and values ❷. When a variable is assigned a new value ❸, the old value is forgotten, which is why `spam` evaluated to 42 instead of 40 at the end of the example. This is called *overwriting* the variable.

BIẾN (VARIABLE)

❑ Khái niệm về biến:

```
>>> spam = 'Hello'
>>> spam
'Hello'
>>> spam = 'Goodbye'
>>> spam
'Goodbye'
```

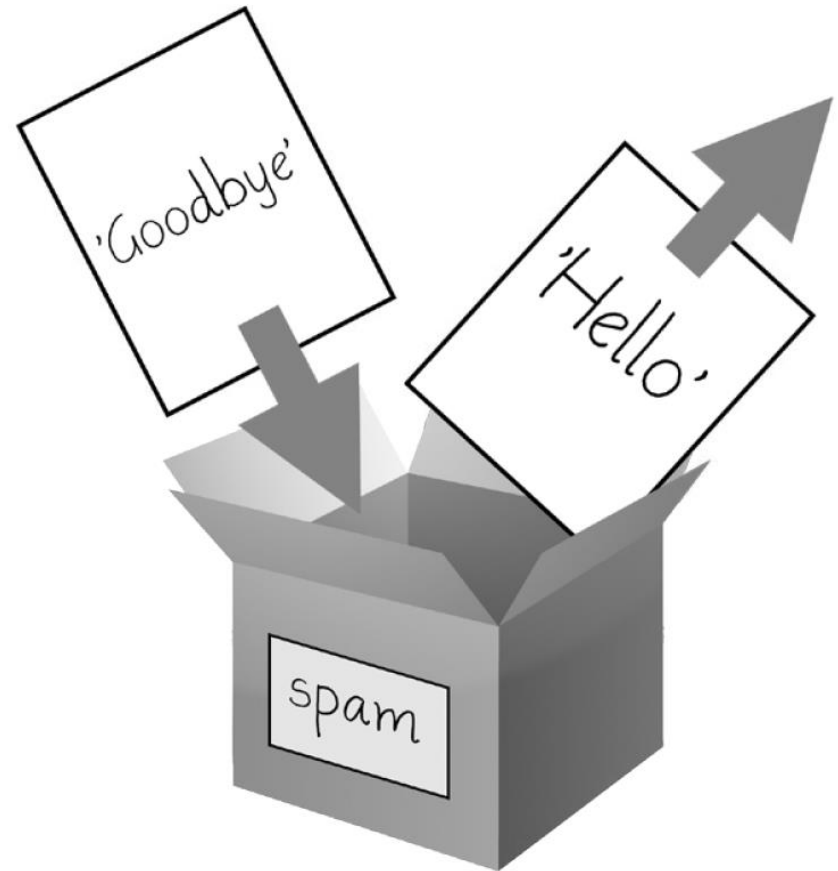


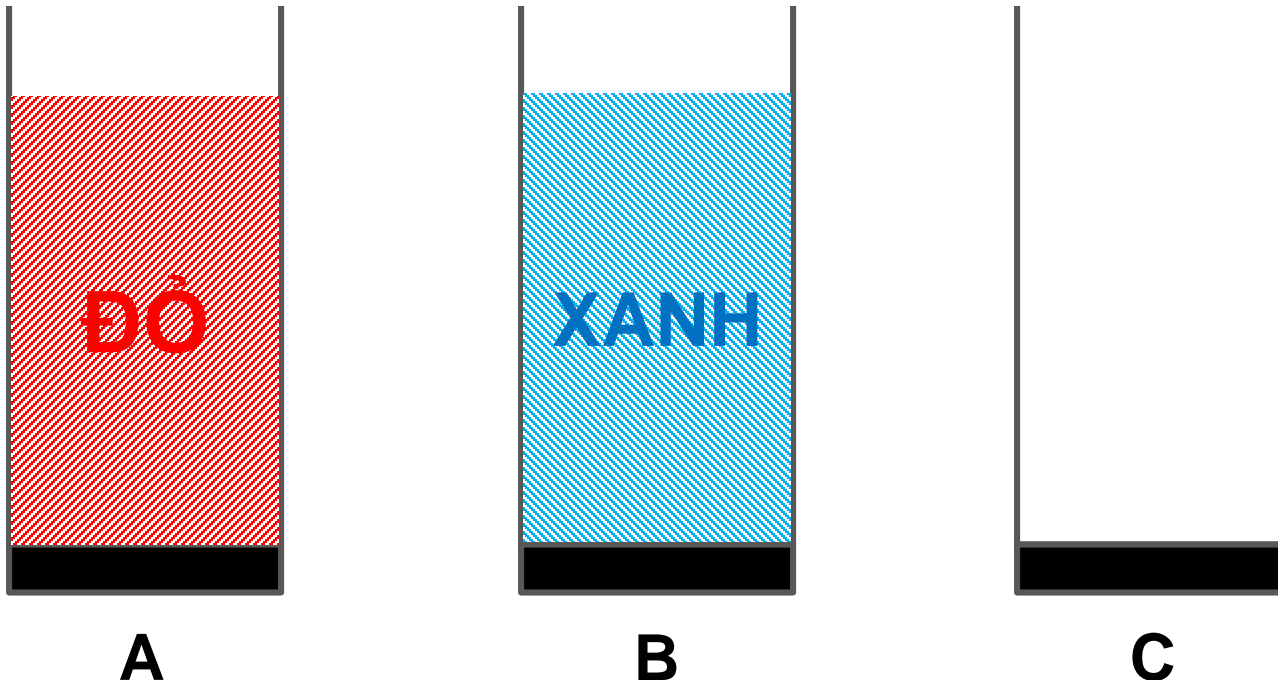
Figure 1-3: When a new value is assigned to a variable, the old one is forgotten.



BIẾN (VARIABLE)

❑ Khái niệm về biến:

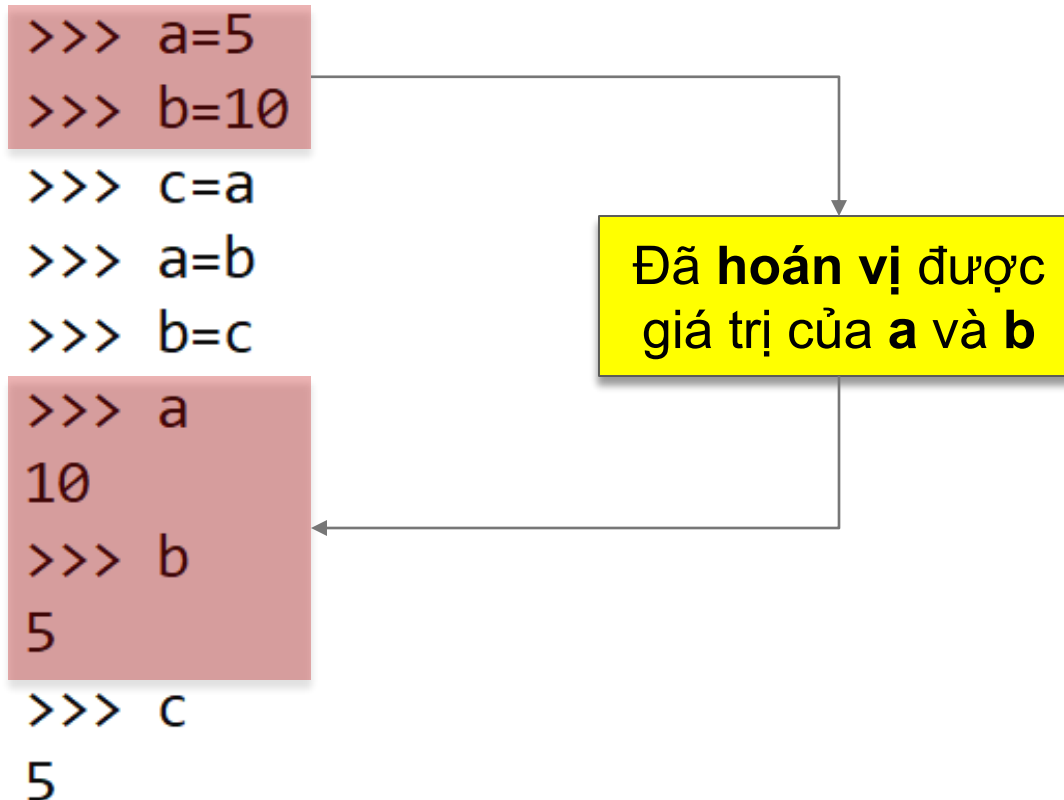
- Ví dụ: sử dụng biến trong bài toán hoán vị.
 - Trình bày phương pháp hoán vị nước giữa hai ly **A** và **B**, sử dụng ly trung gian **C**



BIẾN (VARIABLE)

❑ Khái niệm về biến:

- **Ví dụ:** sử dụng biến trong bài toán hoán vị 3 số nguyên.



ĐỊNH DANH

❑ Quy tắc đặt tên các định danh (identifier)

- Tên của một định danh (*biến, hàm, lớp,...*) là chuỗi gồm các ký tự: **chữ cái** (*chữ hoa hoặc chữ thường*), **chữ số** và **dấu gạch chân** (_);
- Tên **không** được bắt đầu bằng **chữ số**;
- Tên trong python có phân biệt **chữ HOA** và **chữ thường**
 - spam, SPAM, Spam, sPaM: là những tên khác nhau;
- Không được sử dụng **từ khóa** của python để đặt tên;
- Trong Python, thường bắt đầu với ký tự thường và dùng những tên có ý nghĩa và dễ phân biệt;

ĐỊNH DANH

❑ Quy tắc đặt tên các định danh (identifier)

Valid variable names	Invalid variable names
balance	current-balance (hyphens are not allowed)
currentBalance	current balance (spaces are not allowed)
current_balance	4account (can't begin with a number)
_spam	42 (can't begin with a number)
SPAM	total_\$um (special characters like \$ are not allowed)
account4	'hello' (special characters like ' are not allowed)

ĐỊNH DANH

❑ Quy tắc đặt tên cho các định danh

- Ví dụ: xác định tên đúng/ sai trong các trường hợp

TT	Tên biến	Đúng/Sai ?
1	Employee	
2	Else	
3	_Name	
4	Emp_Name	
5	@goto	
6	def	
7	4myclass	
8	Student&Class	

CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

❑ Hàm print()

- Cú pháp: **print**(<chuỗi/ biến / biểu thức>)
- Ý nghĩa: in lên màn hình giá trị một chuỗi, biến hoặc biểu thức;

```
>>> print("Hello World!!!")
Hello World!!!
>>> print(150)
150
>>> a=5
>>> b=10
>>> print(a+b)
15
>>> print("a+b=", a+b)
a+b= 15
>>> print(a,b,a+b)
5 10 15
```

CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

❑ Hàm print()

- Viết không xuống dòng

print(<chuỗi/ biến / biểu thức>, end=<string>)

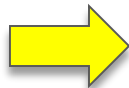
a=10

b=20

a=a+b

print(a)

print(b)



30

20

a=20

b=20

a=a+b

print(a, end=" ")

print(b)



30 20

CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

❑ Hàm input()

- **Cú pháp:** `<Biến> = input([<Chuỗi thông báo>])`
- **Ý nghĩa:** chờ nhập từ bàn phím một chuỗi ký tự, kết thúc nhập bằng phím enter. Giá trị được nhập sẽ được gán vào `<Biến>`;

```
>>> Name=input()
```

```
Co so lap trinh
```

```
>>> print(Name)
```

```
Co so lap trinh
```

Biến **Name** có giá trị được nhập từ bàn phím

```
>>> Name=input("Nhap ten mon: ")
```

Chuỗi thông báo

```
Nhap ten mon: Co so lap trinh
```

```
>>> print("Ten mon: ",Name)
```

```
Ten mon: Co so lap trinh
```

```
>>> print("Ten mon: " + Name)
```

```
Ten mon: Co so lap trinh
```

Hai cách viết tương đương

CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

❑ Hàm len()

- Cú pháp: **len**(<chuỗi>)
- Ý nghĩa: tính và trả về chiều dài của <chuỗi>;

```
>>> print(len("Co so lap trinh"))
```

```
15
```

```
>>> print(len("Toi hoc" + " Co so lap trinh"))
```

```
23
```

```
>>> str="Co so lap trinh"
```

```
>>> print(len(str))
```

```
15
```

CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

❑ Các hàm chuyển kiểu dữ liệu

- **str**(<value>): chuyển <value> thành chuỗi;
- **int**(< value >): chuyển <value> thành số nguyên;
- **float**(< value >): chuyển <value> thành số thực;

```
>>> str(29)
'29'
>>> print('I am ' + str(29) + ' years old.')
I am 29 years old.
```

Nguồn: Automate the Boring Stuff with Python

```
>>> print('aa' + str(111) + 'bb')
aa111bb
>>> print('I am ' + 29 + ' years old.')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str, not int
```

Không thể cộng chuỗi với số được

CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

❑ Các hàm chuyển kiểu dữ liệu

```
>>> str(0)
'0'
>>> str(-3.14)
'-3.14'
>>> int('42')
42
>>> int('-99')
-99
>>> int(1.25)
1
>>> int(1.99)
1
>>> float('3.14')
3.14
>>> float(10)
10.0
```

Hàm **int()** còn dùng để lấy **phần nguyên** của một số thực

CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

```
print('You will be ' + str(int(myAge) + 1) + ' in a year.')  
print('You will be ' + str(int( '4' ) + 1) + ' in a year.')  
print('You will be ' + str(    4 + 1    ) + ' in a year.')  
print('You will be ' + str(        5        ) + ' in a year.')  
print('You will be ' +          '5'          + ' in a year.')  
print('You will be 5'                    + ' in a year.')  
print('You will be 5 in a year.')
```

Figure 1-4: The evaluation steps, if 4 was stored in myAge

CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

- **Ví dụ:** viết chương trình python cho phép nhập vào 2 số nguyên a và b, in lên màn hình tổng của 2 số đó.

```
a=int(input('a='))  
b=int(input('b='))  
print('a+b=',a+b)
```



```
a=5  
b=10  
a+b= 15
```


CÁC HÀM NHẬP, XUẤT DỮ LIỆU CƠ BẢN

■ Lời chú thích

- Cú pháp: **#<văn bản>**
- Ý nghĩa: chuyển nội dung <văn bản> thành lời chú thích/ ghi chú.

```
print('Hello world!')  
print('What is your name?') # ask for their name → Lời chú thích/ ghi chú
```

BÀI TẬP ÔN TẬP

Câu 2.1. Viết chương trình thực hiện yêu cầu sau:
Nhập từ bàn phím **Họ tên** và **Tuổi** của bạn
In lên màn hình câu thông báo theo mẫu:

```
Ho ten: An
```

```
Tuoi: 18
```

```
Chao ban An, chuc mung sinh nhat thu 18 !!!
```

BÀI TẬP ÔN TẬP

Câu 2.2. Viết chương trình tính và in lên màn hình **giá bán** của các mặt hàng với công thức sau:

$$\text{Giá bán} = \text{Giá niêm yết} - \text{Chiết khấu} + \text{VAT}$$

Trong đó:

- Nhập từ bàn phím **Giá niêm yết, Chiết khấu**
- **VAT = (Giá niêm yết – Chiết khấu) * 0.01**

```
Nhap Gia niem yet: 100
```

```
Nhap Chiet khau: 10
```

```
Gia ban: 90.9
```

BÀI TẬP ÔN TẬP

Câu 2.3. Viết chương trình để tính và in lên màn hình **Tiền lãi cuối kỳ** tiết kiệm tại ngân hàng, như sau:

- **Tiền vốn đầu tư ban đầu** là **P** đồng;
- **Lãi suất tiền gửi mỗi tháng** là **r**, được lãi vào cuối kỳ;
- Sau **n** tháng, số tiền thu về cả vốn và lãi là:

$$\text{Tiền lãi cuối kỳ} = P * (1 + r * n)$$

Trong đó:

- **P**, **r** và **n** được nhập từ bàn phím

```
Tien dau tu ban dau: 1000
```

```
So thang gui: 10
```

```
Lai suat moi thang: 0.05
```

```
Tien lanh cuoi ky: 1500.0
```

BÀI TẬP ÔN TẬP

Câu 2.4. Viết chương trình tính và in lên màn hình diện tích của một tam giác theo độ dài của các cạnh, theo công thức:

$$\text{Diện tích} = \sqrt{s(s-a)(s-b)(s-c)}$$

Trong đó:

- **a, b** và **c** là độ dài của 3 cạnh tam giác, được nhập từ bàn phím;
- **$S=(a+b+c)/2$**

a=4

b=6

c=5

Dien tich=9.921567416492215

*Gợi ý: Sử dụng lệnh **import math** để khai báo thư viện **math**.
Lệnh **math.sqrt(X)** để tính căn bậc hai của một số **X** bất kỳ.*