

Control Statements - Repetition/Looping Structures

Student's Name

University

Course

Professor

Date

Control Statements - Repetition/Looping Structures

Introduction

Usually, statements in a program are executed line by line in the sequence of their occurrences. This is known as *sequential execution*, as shown in Fig 1 (Deitel & Deitel, 2017). However, some statements enable us to specify that sequential execution can be avoided as the need arises. This is known as the *transfer of control*, as shown in Fig 2 (Deitel & Deitel, 2017).

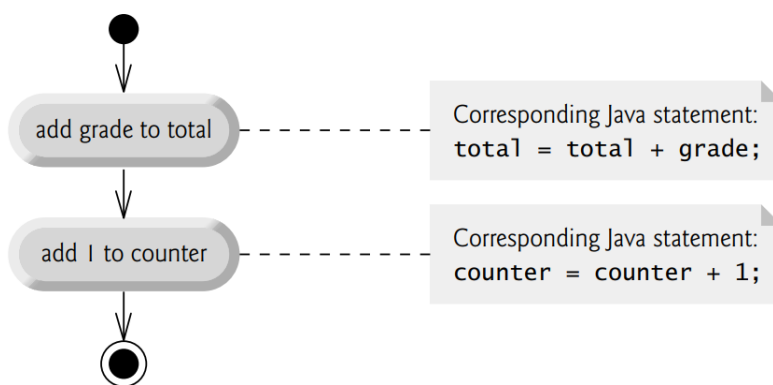


Figure 1: Sequence-structure activity diagram

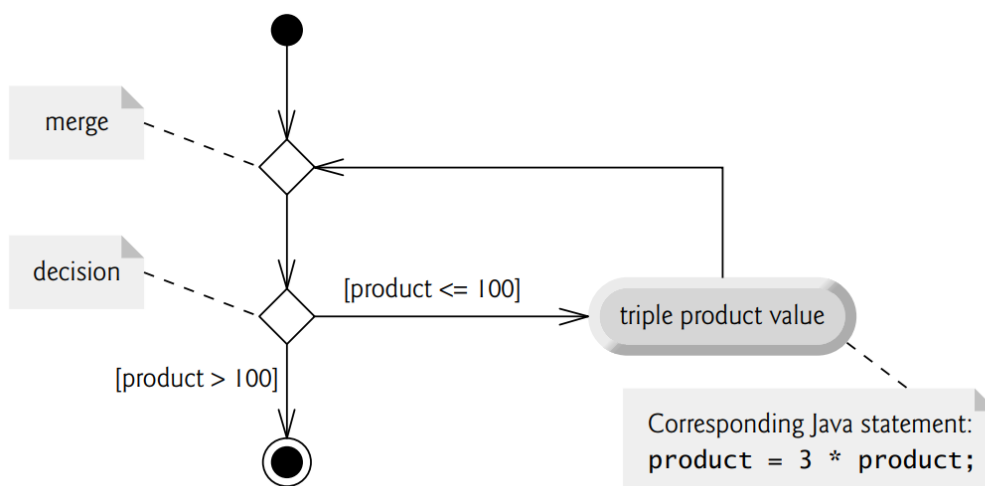


Figure 2: while repetition statement UML activity diagram

In a loop, a part of the program is repeated over and over until a specific goal is achieved. Loops are significant for calculations that require repeated procedures for processing large

data (Horstmann, 2013). For instance, printing odd/even numbers in a list of numbers between 0 and 20 inclusive. Loop implementation differs in every programming language. Java programming language, for example, has three different implementations, that is, *do-while*, *while*, and *for* (Deitel & Deitel, 2017). Others, like python, support only *for* and *while* loops.

Types of Loops

1. do-while loop

The do-while loop executes at least once, and it continues the execution until a given condition is achieved. It is also called a post-tested loop. It is mostly used with menu-driven programs. The do-while loop syntax for a Java program is:

```
do {  
    // code to execute  
} while(condition);
```

2. while loop

Also referred to as a pre-tested loop. It is used in cases where the expected number of iterations is not known from the start. The code block is executed until the condition specified in the loop is satisfied. The while loop syntax for a Java program is:

```
while (condition) {  
    // code to execute  
}
```

3. for loop

It allows for the execution of some parts of the program until the given condition is fulfilled. It is mostly used for a known number of iterations. The for-loop syntax for a Java program is:

```
for (initialization; condition; increment/decrement) {
```

```
        // code to execute  
    }
```

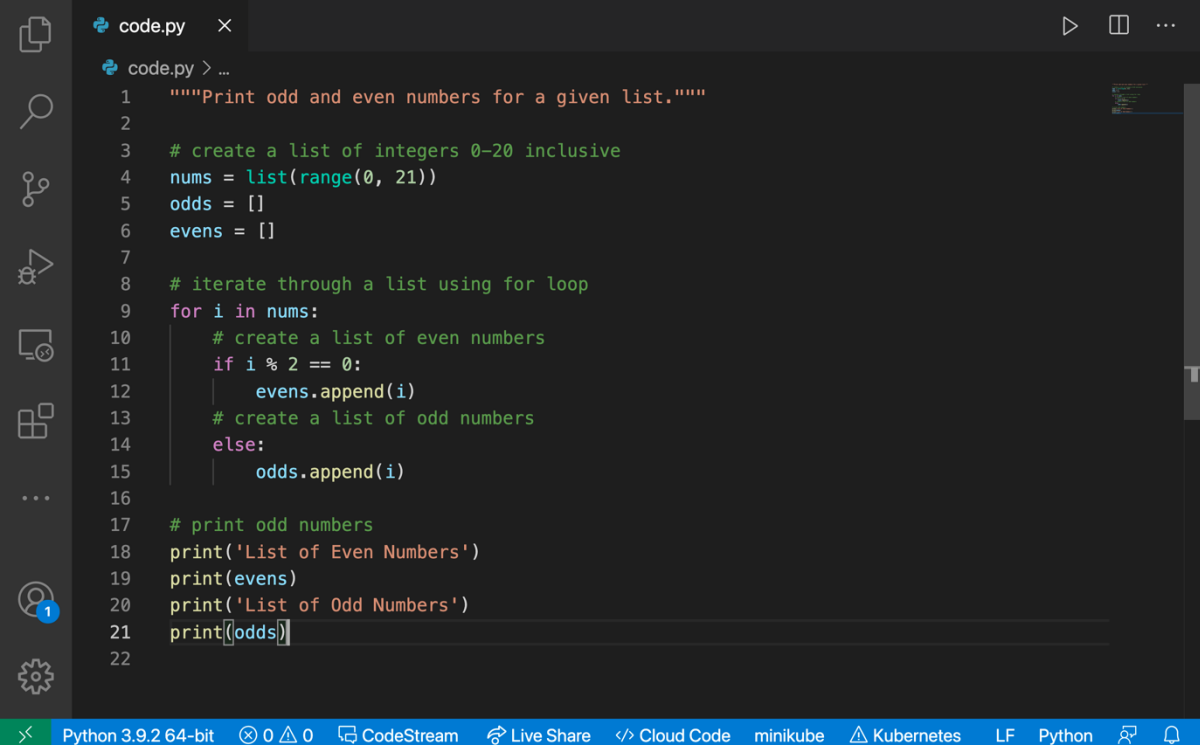
Advantages of loops

1. Provides code reusability.
2. Allows us to avoid code duplication.
3. Loops allow for traversing or iterating over elements of data structures, for instance, arrays or linked lists.

Disadvantages

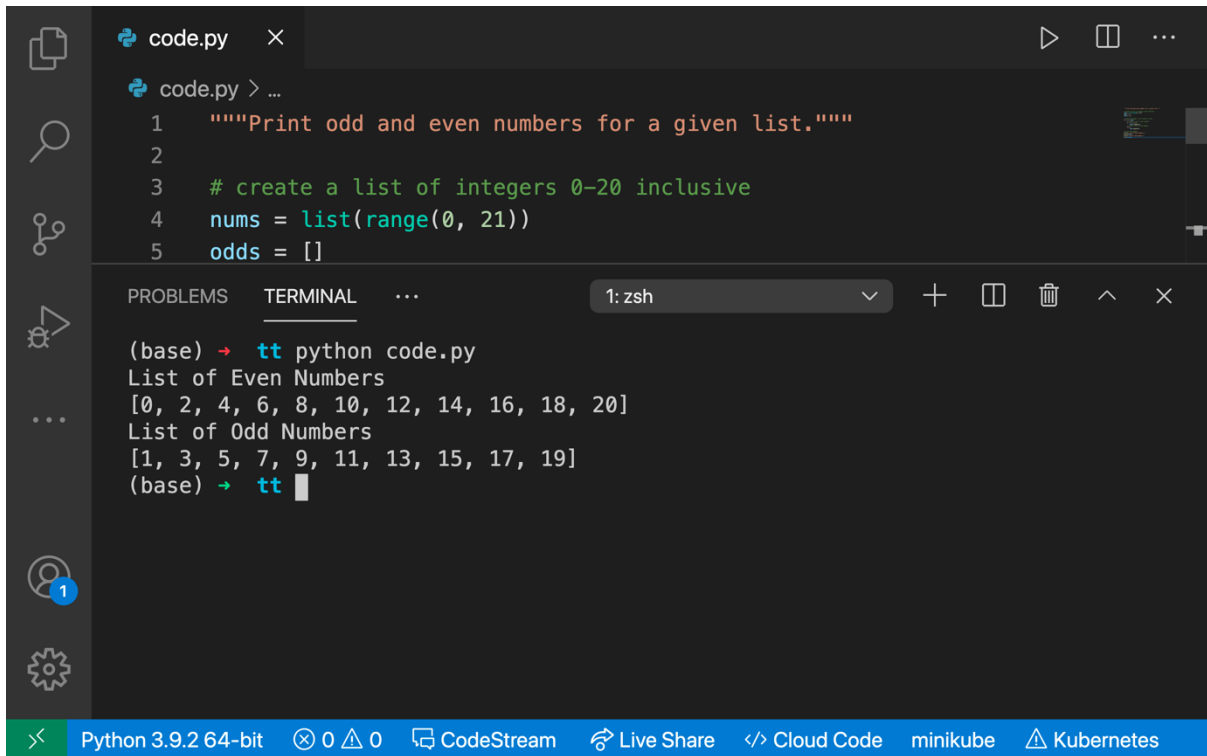
1. The increased time complexity of an algorithm, the Big O notation.

Python Code Sample

A screenshot of a code editor window titled 'code.py'. The code is written in Python and is designed to print odd and even numbers from a list of integers from 0 to 20. The code includes comments explaining each step: creating a list, iterating with a for loop, checking for even/odd numbers, and printing the results. The editor interface shows a dark theme, a search bar on the left, and a status bar at the bottom indicating 'Python 3.9.2 64-bit' and other system information.

```
code.py > ...  
1  """Print odd and even numbers for a given list."""  
2  
3  # create a list of integers 0-20 inclusive  
4  nums = list(range(0, 21))  
5  odds = []  
6  evens = []  
7  
8  # iterate through a list using for loop  
9  for i in nums:  
10     # create a list of even numbers  
11     if i % 2 == 0:  
12         evens.append(i)  
13     # create a list of odd numbers  
14     else:  
15         odds.append(i)  
16  
17 # print odd numbers  
18 print('List of Even Numbers')  
19 print(evens)  
20 print('List of Odd Numbers')  
21 print(odds)  
22
```

Figure 3: Python code sample to print odd and even numbers between 0 and 20



The screenshot shows a code editor with a dark theme. The top part displays a Python script named 'code.py' with the following code:

```
1 """Print odd and even numbers for a given list."""
2
3 # create a list of integers 0-20 inclusive
4 nums = list(range(0, 21))
5 odds = []
```

The bottom part of the editor shows a terminal window with the following output:

```
(base) → tt python code.py
List of Even Numbers
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
List of Odd Numbers
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
(base) → tt
```

The terminal window title is '1: zsh'. The bottom status bar of the editor shows 'Python 3.9.2 64-bit', '0 0', 'CodeStream', 'Live Share', 'Cloud Code', 'minikube', and 'Kubernetes'.

Figure 4: Code output

Figure 3 shows a python code screenshot that iterates through an array (list) and creates other lists of even and odd numbers. Figure 4 is the output of running the python code in Figure 3.

References

Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program, Late Objects* (p. 1248). New York City, NY: Pearson.

Horstmann, C. (2013). *Python for everyone*. Wiley.

Custom Writings.com

Order a custom written paper of high quality



Qualified
writers only



Plagiarism free
guarantee



It'll take you
just 2 minutes

[Order now](#)
