

Programming Principles II Final

* Обязательно

* В этой форме будет записано ваше имя. Введите ваше имя.

1

Iterator consists of which methods ? *

☐ `__next__()`

☐ `__prev__()`

☐ `__follow__()`

☐ `__iter__()`

2

Result of execution of following code will be ? *

```
class MyNumbers:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        x = self.a
        self.a += 1
        return x

myclass = MyNumbers()
myiter = iter(myclass)

next(myiter)
print(next(myiter))
print(next(myiter))
print(next(myiter))
```

- ☐ 1 "\n" 2 "\n" 3
- ☒ 2 "\n" 3 "\n" 4
- ☐ 1 "\n" 2 "\n" 3 "\n" 4

3

The main difference between iterator and iterable is ? *

- ☒ Iterable can not save the state of the iteration, whereas iterator can
- ☐ Iterator can not save the state of the iteration, whereas iterable can

4

Result of execution of following code will be ? *

```
someStr = "Programming Principles II"
next(someStr)
print(next(someStr))
```

- ☐ P "\n" r
- ☐ r
- ☐ "\n" r
- ☐ **TypeError: 'str' object is not an iterator**

5

Which element of a list will return **"False"** ? *

```
def it(ob):
    try:
        iter(ob)
        return True
    except TypeError:
        return False

# Driver Code
for i in [34, [4, 5], (4, 5), {"a":4}, "dfsdf", 4.5]:
    print(i,"is iterable :",it(i))
```

- ☐ 34 and "dfsdf"
- ☐ **{"a":4} and "dfsdf"**
- ☐ {"a":4} and 4.5
- ☐ 34 and 4.5

6

Generator function is a function that uses a **"yield"** keyword. Please, choose True or False *

- ☒ True
- ☐ False

7

In order to import and rename a module "mymodule" we can use the following code: *

- ☐ import mymodule
- ☒ import mymodule as mx
- ☐ from mymodule import mx
- ☐ from mymodule import mymodule as mx

8

In order to work with dates as with date objects we can import a following module ... *

- ☐ date
- ☐ time
- ☐ year
- ☐ **datetime**

9

Result of execution of following code will be ? *

```
import datetime  
  
x = datetime.datetime(2020, 5, 17)  
  
print(x)
```

- ☐ 2020-05-17
- ☐ 2020-17-05 00:00:00
- ☐ **2020-05-17 00:00:00**
- ☐ 2020-17-05

10

In order to work with regular expressions we can use a following module ...

*

- ☐ regex
- ☐ **re**
- ☐ RegEx

11

What does a "**\w**" sequence mean ? *

- ☐ returns a match where the string contains digits (numbers from 0-9)
- ☐ returns a match where the string contains a white space character
- ☐ **returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)**
- ☐ returns a match where the string DOES NOT contain any word characters

12

Result of execution of following code will be ? *

```
import re

txt = "The rain in Spain"
x = re.findall("\w+ai\w+$", txt)
print(x)
```

- ☐ ['rain', 'Spain']
- ☐ ['rain']
- ☐ []
- ☐ ['Spain']

13

Result of execution of following code will be ? *

```
import re

txt = "The rain in Spain"
x = re.search("\w+ai\w+", txt)
print(x)
```

- ☐ <re.Match object; span=(3, 9), match=' rain '>
- ☒ <re.Match object; span=(11, 17), match=' Spain'>
- ☐ rain
- ☐ Spain

14

Result of execution of following code will be ? *

```
import re

txt = "The rain in Spain"
x = re.split("ai", txt)
print(x)
```

15

What meaning do "{}" metacharacters have? *

- ☐ Capture and group
- ☐ A set of characters
- ☐ A dictionary
- ☒ Exactly the specified number of occurrences

16

In order to read a file **demofile.txt** and save it to a variable **f** you will write a code: *

17

Content of a "demofile.txt" is following:

'''

Hello! Welcome to demofile.txt

This file is for testing purposes.

Good Luck!

'''

What will be a result of execution of following code ? *

```
f = open("demofile.txt", "r")
print(f.readline().rstrip())
f.close()
print(f.readline().rstrip())
```

- ☐ Hello! Welcome to demofile.txt "\n" "\n" ValueError: I/O operation on closed file.
- ☒ Hello! Welcome to demofile.txt "\n" ValueError: I/O operation on closed file.
- ☐ ValueError: I/O operation on closed file.
- ☐ Hello! Welcome to demofile.txt

18

In order to check if file is closed which word can we use ? *

☐ isClosed

☐ closed

☐ isclosed

☐ closed()

19

Content of a "demofile.txt" is following:

"""

Hello! Welcome to demofile.txt

This file is for testing purposes.

Good Luck!

"""

What will be a result of execution of following code ? *

```
f = open("demofile.txt", "r")
print(f.readline().rstrip())
f.close()
print(f.closed)
```

- ☐ Hello! Welcome to demofile.txt "\n" True
- ☐ Hello! Welcome to demofile.txt "\n" ValueError: I/O operation on closed file.
- ☐ Hello! Welcome to demofile.txt True
- ☐ Hello! Welcome to demofile.txt True

20

Content of a "demofile.txt" is following:

"""

Hello! Welcome to demofile.txt

This file is for testing purposes.

Good Luck!

"""

What will be a content of the file after execution of following code ?

*

```
f = open("demofile.txt", "w")
```

- ☐ Hello! Welcome to demofile.txt "\n" This file is for testing purposes. "\n" Good Luck!
- ☐ empty file

Этот контент не создавался и не подтверждался корпорацией Майкрософт. Предоставленные вами данные отправляются владельцу формы.



Microsoft Forms