|  |  |  |  |
| --- | --- | --- | --- |
| **Студент:** |  | **Группа:** |  |

**Тема 4. Объектно-ориентированное программирование на Python**

**Тема: Классы и объекты**

**Задание 1**

Сокрытие информации о внутреннем устройстве объекта, при котором вся работа с объектом ведется только через общедоступный интерфейс называется:

1. -: абстракцией
2. -: агрегацией
3. -: полиморфизмом
4. -: декомпозицией
5. -: инкапсуляцией

**Задание 2.**

Объекты obj1 и obj2 являются экземплярами класса MyClass. В результате выполнения каких инструкций на экран будет выведено значение False?

1. -: print(issubclass(A, A))
2. -: print(isinstance(a, type(b)))
3. -: print(dir(a) == dir(b))
4. -: print(a==b)
5. -: print(dir(a) is dir(b))

**Задание 3.**

Что будет выведено на экран в результате выполнения следующей программы:

class MyClass:

a = '123'

def \_\_init\_\_(self, a=0):

self.a = 'None'

mc1 = MyClass(5)

print(mc1.a)

1. -: None
2. -: 123
3. -: 0
4. -: сообщение об ошибке
5. -: 5

**Задание 4.**

Что будет выведено на экран в результате выполнения следующей программы:

class Class1:

a = '1'

def \_\_init\_\_(self, a):

Class1.a = self.a\*a

el = Class1(2)

el2 = Class1(2)

print(el.a-el2.a)

1. -: 11
2. -: 1111
3. -: 111111
4. -: 11111111
5. -: сообщение об ошибке

**Задание 5.**

Имеется класс и объект этого класса

class MyClass:

\_\_val = -5

def \_\_init\_\_(self):

…

def pr(self):

print(self.\_\_val)

a = MyClass()

a.pr()

Какие инструкции можно поставить на место «…», чтобы на экран было выведено значение «5»?

1. -: \_\_val = 5
2. -: self.\_\_val = 5
3. -: a.\_\_val = 5
4. -: MyClass.\_\_val = 5
5. -: MyClass.\_\_val = abs(self.\_\_val)

**Задание 6.**

Что будет выведено на экран в результате выполнения следующей программы:

class MyClass:

def \_\_init\_\_(self):

self.\_\_val = 0

def pr():

print(self.\_\_val)

a = MyClass()

a.\_\_val=10

a.pr()

1. -: 0
2. -: 10
3. -: сообщение об ошибке
4. -: 0 10
5. -: self.\_\_val

**Задание 7.**

Имеется класс и объект этого класса

class MyClass:

@staticmethod

def m1 (p):

print(p)

def \_m2(self, p):

print(p)

@staticmethod

def m3(self, p):

print(p)

def m4(p):

print(p)

def \_\_\_m5(self, p):

print(p)

c = MyClass()

Корректно выполнятся инструкции:

1. -: c.m1(1)
2. -: c.\_m2(1)
3. -: c.m3(1)
4. -: c.m4(1)
5. -: c.\_\_m5(1)

**Задание 8.**

Что будет выведено на экран в результате выполнения следующей программы:

class MyClass():

var = 10

def \_\_init\_\_(self, val=0):

self.var = val

c = MyClass(-2)

d = MyClass(c)

print(d.var)

1. -: 0
2. -: 10
3. -: -2
4. -: <\_\_main\_\_.MyClass object…>
5. -: ошибка

**Задание 9.**

Имеется класс

class MyClass:

def add(self, p, n):

return p - n

def pr(self, x):

…

print(a)

o = MyClass()

o.pr(1)

Чтобы вычислить значение a с помощью метода add, нужно использовать оператор …

1. -: a = self.add(x, 2)
2. -: a = MyClass.add(x, 2)
3. -: a = \_MyClass\_\_add(x, 2)
4. -: a = self.add(x)
5. -: a = MyClass.add(x)

**Задание 10.**

Для метода \_\_init\_\_ справедливы следующие утверждения. Метод:

1. -: может иметь параметры кроме self
2. -: вызывается автоматически при создании класса
3. -: часто называют конструктором по аналогии с другими языками
4. -: всегда имеется в классе
5. -: обычно используется для присваивания начальных значений атрибутам экземпляра класса

**Задание 11.**

Имеется класс и объект этого класса

class MyClass:

def met1(self, param):

return param \* 2

def met2 (param):

return param \* 2

@staticmethod

def met3(param):

return param \* 2

def \_met4(self, param):

return param \* 2

def \_\_met5(self, param):

return param \* 2

o = MyClass()

Инструкция o.METHOD(2) корректно выполнится для методов (если подставить название метода вместо слова «METHOD»):

1. -: met1
2. -: met2
3. -: met3
4. -: \_met4
5. -: \_\_met5

**Задание 12.**

Что будет выведено на экран в результате выполнения следующей программы:

class Class1():

var = 10

def \_\_init\_\_(self, val=0):

self.var = val

c = Class1()

d=str(type(c)).split('.')[-1][:-2]

print(d)

|  |  |
| --- | --- |
| Ответ:\_ |  |

**Задание 13.**

После выполнения операторов

class A:

def \_\_init\_\_(self):

self.a = 1

self.b = 10

def pr(self):

print(self.a, self.b)

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.a = 2

el1 = B()

el1.pr()

на экран будет выведено …

1. -: 1 10
2. -: 2 10
3. -: 1 None
4. -: 2 None
5. -: сообщение об ошибке

**Задание 14.**

Укажите количество общедоступных методов в объекте-экземпляре класса ABC и значение, которое будет получено при вызове метода a() для следующей программы:

class A(object):

def a(self): return 'a'

class B(object):

def b(self): return 'b'

class C(object):

def c(self): return 'c'

class AB(A, B):

def a(self): return 'ab'

class BC(B, C):

def a(self): return 'bc'

class ABC(AB, B, C):

def a(self): return 'abc'

1. -: 3, возвратит ab
2. -: 1, возвратит abc
3. -: 5, возвратит abc
4. -: 3, возвратит abc
5. -: 1, возвратит a

**Задание 15.**

Как называется отношение, которое имеют между собой следующие два класса:

class A(type):

def \_\_init\_\_(cls, name, bases, dict):

return super(A, cls).\_\_init\_\_(cls, name, bases, dict)

B = A("B", (), {})

1. -: ассоциация. A содержит ссылки на B
2. -: наследование. B получается наследованием A
3. -: метакласс. A является метаклассом для B
4. -: метакласс. B является метаклассом для A
5. -: наследование. A получается наследованием B

**Задание 16.**

Имеется следующая программа

class A:

def \_\_init\_\_(self):

self.\_\_a = 1

self.b = 10

def pr(self):

print(self.\_\_a, self.b)

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_()

…

obj1 = B()

obj1.pr()

Чтобы изменить значение атрибута \_\_a в классе A из конструктора производного класса, нужно вместо многоточия в методе \_\_init\_\_() написать оператор …

1. -: self.\_\_a = 5
2. -: A\_\_a = 5
3. -: \_B\_\_a = 5
4. -: self.\_A\_\_a = 5
5. -: super().\_\_a = 5

**Задание 17.**

Даны следующие классы:

class Table:

def \_\_init\_\_(self, l, w, h=10):

self.length = l

self.width = w

self.height = h

class KitchenTable(Table):

def setPlaces(self, p):

self.places = p

class DeskTable(Table):

def square(self):

return self.width \* self.length

Укажите инструкции, которые исполнятся корректно

1. -: t1 = KitchenTable(10,10)
2. -: t2 = KitchenTable(5)
3. -: t3 = DeskTable(\*(1,2))
4. -: t4 = KitchenTable(1, 2, 3)
5. -: t5 = DeskTable(0)

**Задание 18.**

Напишите строчку, которая объявляет класс A, родителем которого является класс B (напишите ответ на английском языке в синтаксисе Python, включая знаки препинания и используя только один пробел)

|  |  |
| --- | --- |
| Ответ:\_ |  |