

ОТЧЕТ

ПРАКТИЧЕСКАЯ РАБОТА №17

ТЕМА: Составление программ с помощью ООП

ЦЕЛЬ: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community

Постановка задачи:

- 1) Создайте класс «Матрица», который имеет атрибуты количества строк и столбцов. Добавьте методы для сложения, вычитания и умножения матриц.
- 2) Создание базового класса "Фигура" и его наследование для создания классов "Квадрат", "Прямоугольник" и "Круг". Класс "Фигура" будет иметь общие методы, такие как вычисление площади и периметра, а классы-наследники будут иметь специфичные методы и свойства.

Текст программы (1):

```
import random
class Matrix:
    def __init__(self, stroks, stolbs):
        self.stroks = stroks
        self.stolbs = stolbs

        self.matrix_1 = [[random.randint(1, 30) for i in
range(self.stolbs)] for k in range(self.stroks)]
        self.matrix_2 = [[random.randint(1, 30) for i in
range(self.stolbs)] for k in range(self.stroks)]

    def addition_matrix(self):
        answer_matrix = []
        itogo_matrix = []
        for i in range(len(self.matrix_1)):
            for k in range(len(self.matrix_1[0])):

answer_matrix.append(self.matrix_1[i][k] +
self.matrix_2[i][k])
                while len(answer_matrix) > 0:
```

```

        itogo_matrix.append([i for i in
answer_matrix[:len(answer_matrix)]])
        del answer_matrix[:self.stolbs-1]
        if len(answer_matrix) == 0:
            break
    return itogo_matrix

def subtraction_matrix(self):
    answer_matrix = []
    itogo_matrix = []
    for i in range(len(self.matrix_1)):
        for k in range(len(self.matrix_1)):
            answer_matrix.append(self.matrix_1[i][k] -
self.matrix_2[i][k])
            while len(answer_matrix) > 0:
                itogo_matrix.append([i for i in
answer_matrix[:self.stolbs]])
                del answer_matrix[:self.stolbs]
                if len(answer_matrix) == 0:
                    break
    return itogo_matrix

def multiple_matrix(self):
    if self.stolbs == self.stroks:
        itogo_matrix = [[0 for k in
range(self.stolbs)] for i in range(self.stroks)]
        for i in range(len(self.matrix_1)):
            for k in range(len(self.matrix_2[0])):
                for j in range(len(self.matrix_2)):
                    itogo_matrix[i][k] += self.matrix_1[i][j]
* self.matrix_2[j][k]
        return itogo_matrix
    else:
        return "Построение невозможно"

ob = Matrix(4,4)
print(ob.matrix_1)
print(ob.matrix_2)
print()
print(ob.addition_matrix())

```

```
print(ob.subtraction_matrix())
print(ob.multiple_matrix())
```

Текст программы (2):

```
class Figure:
    def __init__(self, a, b, flag):
        self.a = a
        self.b = b
        self.flag = flag

    def plosh(self):
        if self.flag == 1:
            return self.a * self.b
        elif self.flag == 2:
            return self.a**2
        else:
            return 3.14*(self.a**2)

    def perimetr(self):
        if self.flag == 1:
            return (self.a*2) + (self.b*2)
        elif self.flag == 2:
            return self.a*4
        else:
            return 3.14*2*self.a

class Square(Figure):
    def __init__(self, a, flag):
        self.a = a
        self.flag = flag
    def print(self):
        return "Я квадрат"

class Prom(Figure):
    def __init__(self, a, b, flag):
        self.a = a
        self.b = b
        self.flag = flag
    def print(self):
```

```

        return "Я прямоугольник"

class Circle(Figure):
    def __init__(self, a, flag):
        self.a = a
        self.flag = flag
    def print(self):
        return "Я круг"

ob = Square(3,2)
print(ob.plosh())
print(ob.perimetr())
print(ob.print())

print()

ob2 = Prom(10, 25, 1)
print(ob2.plosh())
print(ob2.perimetr())
print(ob2.print())

print()

ob3 = Circle(15,3)
print(ob3.plosh())
print(ob3.perimetr())
print(ob3.print())

```

Протокол работы:

1)

```

[[2, 4, 18, 8], [23, 29, 16, 26], [30, 11, 2, 9], [22, 1, 12, 10]]
[[9, 6, 5, 15], [6, 8, 7, 25], [19, 15, 21, 18], [27, 29, 30, 11]]

```

```

[[11, 10, 23, 23], [29, 37, 23, 51], [49, 26, 23, 27], [49, 30, 42, 21]]
[[-7, -2, 13, -7], [17, 21, 9, 1], [11, -4, -19, -9], [-5, -28, -18, -1]]
[[600, 546, 656, 542], [1387, 1364, 1434, 1644], [617, 559, 539, 860],
[702, 610, 669, 681]]

```

2)

9

12

Я квадрат

250

70

Я прямоугольник

706.5

94.2

Я круг

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с использованием ООП. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.