



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №5

«Объектно-ориентированное программирование на Python»

ДИСЦИПЛИНА: «Высокоуровневое программирование»

Выполнил: студент гр. ИУК4-32Б _____ (Подпись) (Карельский М.К.)
(Ф.И.О.)

Проверил: _____ (Подпись) (Ерохин И.И.)
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга , 2021

Цель: формирование практических навыков процедурного программирования, разработки и отладки программ, овладение методами и средствами разработки и оформления технической документации.

Вариант 10

Задание:

Задача 1.

Создайте класс `Peoples`, задайте ФИО. Создайте класс-наследник `Students`, заполните его ФИО. Создайте для него метод `ChangeAstimate` для изменения оценок студента, метод `WriteAstimate` – для вывода массива с оценками и ФИО. Задайте произвольные ФИО и оценки. Выведите список, состоящий из трёх студентов.

Задача 2.

Написать программу с использованием абстрактного класса для нахождения среднего арифметического заданного ряда целых чисел.

Задача 3.

Создать класс `Password` и описать в нем ник человека на сайте и его пароль, при этом пароль может быть четырёхзначным пин-кодом (целое число, состоящее из 4 цифр) или полноценным паролем (строкой). Описать метод `coincidence`, определяющий, является ли введённое с клавиатуры `n` паролем для данного ника.

Задача 4.

Создать абстрактный базовый класс «`Triangle`» для треугольника и вычислить площадь и периметр с помощью абстрактных методов. Поля данных должны включать две стороны и угол между ними. Определить производные классы: прямоугольный треугольник, равнобедренный треугольник, равносторонний треугольник с собственными функциями вычисления площади и периметра.

Листинг:

Задача 1

```
class People:
    def __init__(self):
        self.names = [str]

    names = [str]

class Students(People):
    def __init__(self):
        self.estimates = [int]
    def ChangeEstimate(self, index, estimate):
        if estimate > 0 and estimate < 6:
            self.estimate[index] = estimate
    def WriteEstimates(self):
        print('\n'.join("{}: {}".format(self.names[index],
self.estimate[index]) for index in range(len(self.names))))
```

```

    estimates = [int]

arr = Students()
arr.names = ["Kate", "Bob", "Mike"]
arr.estimates = [5, 3, 4]
arr.WriteEstimates()

```

Задача 2

```

from abc import ABC, abstractmethod

class Calculate(ABC):
    @abstractmethod
    def Calculate(self):
        pass

class ArithmeticMean(Calculate):
    def __init__(self, row):
        self.row = row
    def Calculate(self):
        return sum(self.row) / len(self.row)

row = [int]

mean = ArithmeticMean([1, 2, 6])
print(mean.Calculate())

```

Задача 3

```

class Password():
    def __init__(self, nickname, pin, password):
        self.nickname = nickname
        self.pin = pin
        self.password = password
    def Coincidence(self, password):
        return password == self.pin or password == self.password

nickname = str
pin = int
password = str

password = Password("Bob", 1234, "qwerty")
print(password.Coincidence(1234))
print(password.Coincidence("qwerty"))
print(password.Coincidence(0))

```

Задача 4

```

from abc import ABC, abstractmethod
import math

class Triangle:
    @abstractmethod

```

```

def FindPerimeter(self):
    pass
@abstractmethod
def FindSquare(self):
    pass

firstSide = float
secondSide = float
angle = float

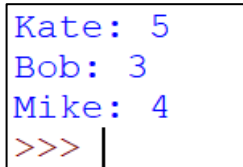
class RectangularTriangle:
    def FindPerimeter(self):
        return firstSide + secondSide + math.tan(angle) *
secondSide
    def FindSquare(self):
        return 0.5 * secondSide * math.tan(angle) * secondSide

class IsoscelesTriangle:
    def FindPerimeter(self):
        return 2 * firstSide + secondSide
    def FindSquare(self):
        return 0.5 * firstSide * math.sin(angle) + secondSide

class EquilateralTriangle:
    def FindPerimeter(self):
        return 3 * firstSide
    def FindSquare(self):
        return firstSide**2 * 3**0.5 / 4

```

Результаты:

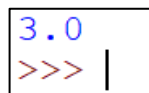


```

Kate: 5
Bob: 3
Mike: 4
>>> |

```

Рисунок 1. Задача 1

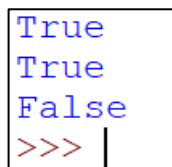


```

3.0
>>> |

```

Рисунок 2. Задача 2



```

True
True
False
>>> |

```

Рисунок 3. Задача 3

Вывод: в ходе выполнения лабораторной работы были получены навыки объектно-ориентированного программирования на Python, создания конструкторов, наследников, абстрактных классов и методов.