

Практическое задание №16

Тема: составление программ с использованием ООП в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ с использованием ООП в IDE PyCharm Community.

Постановка задачи №1:

```
# Создайте класс «Студент», который имеет  
атрибуты имя, фамилия и оценки.  
  
# Добавьте методы для вычисления среднего  
балла и определения, является ли студент  
# отличником.
```

Тип алгоритма: линейный

```
class student:

    def init (self, name,
sec_n, marks):
        self.name = name
        self.sec_n = sec_n
        self.marks = marks

    @staticmethod
    def ave_marks(marks):
        average = sum(marks) /
len(marks)
        return average

    @staticmethod def
    res(average):
        return 4.5 <= average <= 5

Vitaliy = student('Виталий', 'Savage',
[5, 5, 5, 5, 4, 5, 3])

print(Vitaliy.ave_marks(Vitaliy.marks))
print('Student is otlichnik -',
Vitaliy.res(Vitaliy.ave_marks(Vitaliy.m
arks)))
```

Протокол работы программы:

4.571428571428571

Student is otlichnik - True

Process finished with exit code 0

```
# Создайте класс "Человек", который содержит информацию о имени, возрасте и поле.  
# Создайте классы "Мужчина" и "Женщина", которые наследуются от класса "Человек". Каждый класс должен иметь метод, который выводит информацию о поле объекта.
```

Постановка задачи №2:

Тип алгоритма: линейный

Текст программы:

```
class human:  
    def init      (self, name, age, sex): #  
        method self.name = name  
self.age = age self.sex =  
        sex  
  
class man(human):  
    def init      (self, name, age, sex): #  
        method super().      init (name,  
        age, sex)  
  
class woman(human):  
    def init      (self, name, age, sex): #  
        method super().      init (name,  
        age, sex)  
  
Vitaliy = man('Vitek', '25', 'Мужчина')  
print("Пол человека -", Vitaliy.sex)
```

Протокол работы программы:

Пол человека - Мужчина

Process finished with exit code 0

Постановка задачи №3:

Тип алгоритма: линейный

Текст программы:

```
# Для задачи из блока 1 создать две функции, save_def и load_def, которые
# позволяют
# сохранять информацию из экземпляров класса (3 шт.) в файл и загружать ее
# обратно.
# Использовать модуль pickle для сериализации и десериализации объектов Python
# в
# бинарном формате.

import pickle

class student:

    def __init__(self, name, sec_n, marks):
        self.name = name
        self.sec_n = sec_n
        self.marks = marks

    @staticmethod
    def ave_marks(marks):
        average = sum(marks) / len(marks)
        return average

    @staticmethod
    def res(average):
        st_is_otl = 4.5 <= average <= 5
        return st_is_otl

def save_def(stude, file):
    with open(file, 'wb') as f: # wb - запись в бинарном ф
        pickle.dump(stude, f) # запись в файл f
```

```

def load_def(file):
    with open(file, 'rb') as f: # rb - чтение в 2ичном ф
        studen = pickle.load(f)
    return studen

Kovalkin= student('Артём', 'Ковалкин', [2, 5, 5, 3, 4, 5, 4])
Vitebsk = student('Денис', 'Витебск', [3, 2, 5, 4, 4, 5, 5])
KioRio = student('Стас', 'КиoРиo', [5, 4, 5, 4])

students_info = [Kovalkin,Vitebsk,KioRio]

for stud in students_info:
    save_def(stud, 'students.pkl')
    students = load_def('students.pkl')
    print(students.ave_marks(students.marks))
    print('Student is otlichnik -',
students.res(students.ave_marks(students.marks)))

```

Протокол работы программы:

4.0

Student is otlichnik - False

4.0

Student is otlichnik - False

4.5

Student is otlichnik - True

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ с использованием ООП в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub

