

Практическое задание №16

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

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки 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

Текст программы:

```
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 respect(average):
        st_is_otl = average == 5
        return st_is_otl
```

```
Bondarev = student('Сергей', 'Бондарев', [2, 1, 2, 3, 4, 5, 3])
```

```
print(Bondarev.ave_marks(Bondarev.marks))
print('Student is otlichnik -', Bondarev.respect(Bondarev.ave_marks(Bondarev.marks)))
```

Протокол работы программы:

2.857142857142857

Student is otlichnik - False

Process finished with exit code 0

Постановка задачи №2:

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.

Тип алгоритма: линейный

Текст программы:

```
class human:
    def __init__(self, name, age, sex):
        self.name = name
        self.age = age
        self.sex = sex

class man(human):
    def __init__(self, name, age, sex):
        super().__init__(name, age, sex)

class woman(human):
    def __init__(self, name, age, sex):
        super().__init__(name, age, sex)

Bondarev = man('Сергей', '18', 'Мужчина')

print("Пол человека -", Bondarev.sex)
```

Протокол работы программы:

Пол человека - Мужчина

Process finished with exit code 0

Постановка задачи №3:

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.

Тип алгоритма: линейный

Текст программы:

```
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 respect(average):
        st_is_otl = average == 5
        return st_is_otl
```

```
def save_def(stude, file):
    with open(file, 'wb') as f:
        pickle.dump(stude, f)
```

```

def load_def(file):
    with open(file, 'rb') as f:
        studen = pickle.load(f)
    return studen

Bondarev = student('Сергей', 'Бондарев', [2, 1, 2, 3, 4, 5, 4])
Korennoi = student('Никита', 'Коренной', [2, 1, 5, 3, 4, 5, 5])
Borzistiy = student('Максим', 'Борзистый', [5, 5, 5, 5])

students_info = [Bondarev, Borzistiy, Korennoi]

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.respect(students.ave_marks(students.marks)))

```

Протокол работы программы:

```

3.0
Student is otlichnik - False
5.0
Student is otlichnik - True
3.5714285714285716
Student is otlichnik - False

Process finished with exit code 0

```

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

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

Готовые программные коды выложены на GitHub.