

## Практическое занятие № 16

**Тема:** Наименование практического занятия: составление программ с использованием ООП. в IDE PyCharm Community

**Цель:** закрепить усвоенные знания, понятия, с использованием ООП в IDE PyCharm Community.

Постановка задачи.

```
#Создайте класс "Товар" с атрибутами "название", "цена" и "количество".  
Напишите  
#метод, который выводит информацию о товаре в формате "Название: название,  
#Цена: цена, Количество: кол-во".
```

```
#Создайте базовый класс "Фигура" со свойствами "ширина" и "высота". От этого  
#класса унаследуйте классы "Прямоугольник" и "Квадрат". Для класса "Квадрат"  
#переопределите методы, связанные с вычислением площади и периметра.
```

```
#Создайте класс "Товар" с атрибутами "название", "цена" и "количество".  
Напишите  
#метод, который выводит информацию о товаре в формате "Название: название,  
#Цена: цена, Количество: кол-во".  
  
#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, Линейный, Линейный

**Текст программы:**

```
class tovar:  
  
    def __init__(self, nazvanie, price, kolvo):  
        self.nazvanie = nazvanie  
        self.price = price  
        self.kolvo = kolvo  
    def infovivot(self):  
        print(f"Название: {self.nazvanie}, Цена: {self.price}, Количество:  
{self.kolvo}")  
  
tovar = tovar("Айфон", 100000, 5)  
  
tovar.infovivot()
```

```
class figura:  
    def __init__(self, width, height):  
        self.width = width  
        self.height = height
```

```

    def ploshad(self):
        return self.width * self.height

    defimetr(self):
        return 2 * (self.width + self.height)

class rectangle(figura):
    pass

class square(figura):
    def __init__(self, side):
        super().__init__(side, side)

    def ploshad(self):
        return self.width ** 2

    defimetr(self):
        return 4 * self.width

ширина_прямоугольника = int(input("Введите ширину прямоугольника: "))
высота_прямоугольника = int(input("Введите высоту прямоугольника: "))
сторона_квадрата = int(input("Введите сторону квадрата: "))

rectangle = rectangle(ширина_прямоугольника, высота_прямоугольника)
square = square(сторона_квадрата)

print(f"Площадь прямоугольника: {rectangle.ploshad()}")
print(f"Периметр прямоугольника: {rectangle.imetr()}")
print(f"Площадь квадрата: {square.ploshad()}")
print(f"Периметр квадрата: {square.imetr()}")

```

```

import pickle

class product:
    def __init__(self, nazvanie, price, kolvo):
        self.nazvanie = nazvanie
        self.price = price
        self.kolvo = kolvo

    def display_info(self):
        print(f"Название: {self.nazvanie}, Цена: {self.price}, Количество: {self.kolvo}")

def save_def(obj_list, filename):
    with open(filename, 'wb') as f:
        pickle.dump(obj_list, f)

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

product1 = product("Айфон", 100000, 5)
product2 = product("Самсунг", 80000, 10)
product3 = product("Нокия", 50000, 15)

save_def([product1, product2, product3], "tovars.pickle")

tovars_list = load_def("tovars.pickle")

```

```
for tovar in tovars_list:  
    tovar.display_info()
```

---

Протокол работы программы:

**Вывод:** Я закрепил знания, понятия, работы с ООП в IDE PyCharm Community.