

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 ООП в IDE PyCharm Community.

PZ_16_1:

Постановка задачи:

11. 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".

Тип алгоритма: линейный

Текст программы:

```
"""
11. 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".
"""

class Product:
    def __init__(self, name: str, cost: float, quantity: int):
        self.name = name
        self.cost = cost
        self.quantity = quantity

    def get_product(self):
        print(
            f"""
Название: {self.name},
Цена: {self.cost},
Количество: {self.quantity}
            """
        )

if __name__ == "__main__":
    arbuз = Product("арбуз", 128.31, 12)
    arbuз.get_product()
```

Протокол работы программы:

Название: арбуз,

Цена: 128.31,

Количество: 12

PZ_16_2:

Постановка задачи:

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.

Тип алгоритма: циклический

Текст программы:

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

    def calculate_square(self):
        print(f"Площадь фигуры: {self.width * self.height}")

    def calculate_perimeter(self):
        print((f"Периметр фигуры: {(self.width + self.height) * 2}"))

class Rectangle(Figure):
    ...

class Quadrant(Figure):
    @staticmethod
    def calculate_square_quadrant(side):
        print(f"Площадь квадрата: {side ** 2}")

    @staticmethod
    def calculate_perimeter_quadrant(side):
        print(f"Периметр: {side * 4}")
    # def calculate_square_quadrant(self):
    #     print(f"Площадь квадрата: {self.width ** 2}")
    #
    # def calculate_perimeter_quadrant(self):
    #     print(f"Периметр: {self.width * 4}")

bob = Rectangle(14, 25)
bob.calculate_square()
bob.calculate_perimeter()

rostov = Quadrant(5, 5)
rostov.calculate_square_quadrant(5)
rostov.calculate_perimeter_quadrant(5)
```

Протокол работы программы:

Площадь фигуры: 350

Периметр фигуры: 78

Площадь квадрата: 25

Периметр квадрата: 20

PZ_16_3:

Постановка задачи:

Для задачи из блока 1 создать две функции, save def 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: циклический

Текст программы:

```
import pickle

from PZ_16.PZ_16_1 import Product

products = [
    Product(name="мяч", cost=2699.99, quantity=9),
    Product(name="монитор", cost=9000, quantity=4),
    Product(name="смартфон", cost=11499, quantity=5)
]

def dump_objs(objs: list):
    for i in range(len(objs)):
        with open(f"out{i}.bin", "wb") as file:
            pickle.dump(objs[i], file)

def load_objs(objs_len: int):
    for i in range(objs_len):
        with open(f"out{i}.bin", "rb") as file:
            products = pickle.load(file)
            print(products.__dict__)

dump_objs(products)
load_objs(len(products))
```

Протокол работы программы:

{'name': 'мяч', 'cost': 2699.99, 'quantity': 9}

{'name': 'монитор', 'cost': 9000, 'quantity': 4}

{'name': 'смартфон', 'cost': 11499, 'quantity': 5}

Вывод: закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ с ООП в IDE PyCharm Community.