

Практическое занятие №16

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

Цели: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community.

Постановка задачи:

Создайте класс «Календарь», который имеет атрибуты год, месяц и день. Добавьте методы для определения дня недели, проверки на високосный год и определения количества дней в месяце.

Текст программы:

```
import pickle
from calendar import monthrange, weekday, isleap
4 usages
class Calendar:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    2 usages (1 dynamic)
    def day_week(self):
        return weekday(self.year, self.month, self.day)

    2 usages (1 dynamic)
    def visokosni(self):
        return isleap(self.year)

    2 usages (1 dynamic)
    def days_month(self):
        return monthrange(self.year, self.month)[1]

years = int(input("Введите год: "))
month = int(input("Введите месяц (1-12): "))
day = int(input("Введите день (1-31): "))

calendar = Calendar(years, month, day)

print("День недели:", calendar.day_week())
print("Високосный год:", calendar.visokosni())
print("Дней в месяце:", calendar.days_month())
```

Протокол программы:

Введите год: 2024
Введите месяц (1-12): 1
Введите день (1-31): 10
День недели: 2
Високосный год: True
Дней в месяце: 31

Process finished with exit code 0

Постановка задачи:

Создайте базовый класс "Животное" со свойствами "вид", "количество лап", "цвет шерсти". От этого класса унаследуйте класс "Собака" и добавьте в него свойства "кличка" и "порода".

Текст программы:

```
class animals:
    def __init__(self, view, count_lap, colors_sherst):
        self.view = view
        self.count_lap = count_lap
        self.colors_sherst = colors_sherst

1 usage
class Dog(animals):
    def __init__(self, view, count_lap, colors_sherst, name, poroda):
        super().__init__(view, count_lap, colors_sherst)
        self.name = name
        self.poroda = poroda

view = input("Введите вид животного: ")
count_lap = int(input("Введите количество лап: "))
colors_sherst = input("Введите цвет шерсти: ")
name = input("Введите кличку собаки: ")
poroda = input("Введите породу собаки: ")

dog = Dog(view, count_lap, colors_sherst, name, poroda)

print("Информация о собаке:")
print("Вид:", dog.view)
print("Количество лап:", dog.count_lap)
print("Цвет шерсти:", dog.colors_sherst)
print("Кличка:", dog.name)
print("Порода:", dog.poroda)
```

Протокол программы:

Введите вид животного: собака
Введите количество лап: 4
Введите цвет шерсти: бежевый
Введите кличку собаки: попугай
Введите породу собаки: такса
Информация о собаке:
Вид: собака
Количество лап: 4
Цвет шерсти: бежевый
Кличка: попугай
Порода: такса

Process finished with exit code 0

Постановка задачи:

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.

Текст программы:

```
def save_def(objects, filename):  
    with open(filename, 'wb') as f:  
        pickle.dump(objects, f)  
  
1 usage  
def load_def(filename):  
    with open(filename, 'rb') as f:  
        return pickle.load(f)  
  
calendar1 = Calendar(year=2023, month=3, day=8)  
calendar2 = Calendar(year=2024, month=2, day=29)  
calendar3 = Calendar(year=2025, month=12, day=31)  
  
save_def(objects=[calendar1, calendar2, calendar3], filename='calendar_data.dat')  
  
loaded_calendars = load_def('calendar_data.dat')  
for calendar in loaded_calendars:  
    print(f"Дата: {calendar.year}-{calendar.month}-{calendar.day}")  
    print(f"День недели: {calendar.day_week()}")  
    print(f"Високосный год: {calendar.visokosni()}")  
    print(f"Количество дней в месяце: {calendar.days_month()}\n")
```

Протокол программы:

Дата: 2023-3-8
День недели: 2
Високосный год: False
Количество дней в месяце: 31

Дата: 2024-2-29
День недели: 3
Високосный год: True
Количество дней в месяце: 29

Дата: 2025-12-31
День недели: 2
Високосный год: False
Количество дней в месяце: 31

Вывод:

Я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навык и составление программ с ООП в IDE PyCharm Community.