

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.

Задача 1

```
# 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.

# 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.
```

Задача 2

```
# Создайте класс "Животное", который содержит информацию о виде и
возрасте

# животного. Создайте классы "Собака" и "Кошка", которые наследуются от
класса

# "Животное" и содержат информацию о породе.
```

Тип алгоритма: линейная, ветвящийся.

Текст программы 1:

```
import datetime
import pickle

class Calendar:
    wd = {0: 'понедельник',
          1: 'вторник',
          2: 'среда',
          3: 'четверг',
          4: 'пятница',
          5: 'суббота',
          6: 'воскресенье'}

    def __init__(self, day, month, year):
        self.year = year
        self.month = month
        self.day = day

    def day_of_week(self):
        print(self.wd[datetime.date(self.year, self.month, self.day).weekday()])

    def days_in_the_month(self):
        month_number = self.month
        year = self.year

        if month_number in (1, 3, 5, 7, 8, 10, 12):
            print('В месяце 31 день')

        elif month_number == 2:
            if year % 4 != 0:
                print('В месяце 28 дней')
            elif year % 100 == 0:
                if year % 400 == 0:
                    print('В месяце 29 дней')
                else:
                    print('В месяце 28 дней')
            else:
                print('В месяце 29 дней')
```

```

def days_in_the_month(self):
    else:
        print('В месяце 30 дней')

def is_visokos(self):
    year = self.year

    if year % 4 != 0:
        print('Год не високосный.')

    elif year % 100 == 0:
        if year % 400 == 0:
            print('Год високосный.')
        else:
            print('Год не високосный.')
    else:
        print('Год високосный.')

def save_def(file, objectDump):
    with open(file, 'wb') as f:
        pickle.dump(objectDump, f)
def load_def(file):
    with open(file, 'rb') as f:
        return pickle.load(f)
try:
    date = int(input('Введите день: '))
    month = int(input('Введите номер месяца: '))
    year = int(input('Введите год: '))
    if date == 0:
        date = 1
    if month == 0:
        month = 1
    if year == 0:
        year = 2024
except:
    print('Введены неверные данные')
print('\n')
calendar = Calendar(date, month, year)
calendar.day_of_week()
calendar.is_visokos()
calendar.days_in_the_month()
print(calendar.__dict__)

save_def('pz16_1_info.bin', calendar.__dict__)
info = load_def('pz16_1_info.bin')
print(info)

```

```

"""
Calendar class
"""
class Calendar:
    """
    Calendar class
    """
    def __init__(self, date, month, year):
        """
        Initialize the calendar
        """
        self.date = date
        self.month = month
        self.year = year
    def day_of_week(self):
        """
        Return the day of the week
        """
        return self.date % 7
    def is_visokos(self):
        """
        Return if the year is a leap year
        """
        return self.is_visokos()
    def days_in_the_month(self):
        """
        Return the number of days in the month
        """
        return self.days_in_the_month()
    def __dict__(self):
        """
        Return the dictionary of the calendar
        """
        return self.__dict__

```

Текст программы 2:

```
class Animal:
    def __init__(self, animal_class, years_old):
        self.who = animal_class
        self.years_old = years_old

class Dog(Animal):
    def __init__(self, who, years_old, breed):
        super().__init__(who, years_old)
        self.breed = breed

class Cat(Animal):
    def __init__(self, animal_class, years_old, breed):
        super().__init__(animal_class, years_old)
        self.breed = breed

try:
    animal_class = input('Введите вид животного: ')
    years_old = int(input('Введите возраст животного: '))
    animal = Animal(animal_class, years_old)
    print('\n', animal.__dict__)

    years_old = int(input('Введите возраст собаки: '))
    breed = input('Введите породу собаки: ')
    dog = Dog('Собака', years_old, breed)
    print('\n', dog.__dict__)

    years_old = int(input('Введите возраст кошки: '))
    breed = input('Введите породу кошки: ')
    cat = Cat('Кот', years_old, breed)
    print('\n', cat.__dict__)
except:
    print('Введены неверные данные')
```

Вывод: закрепил и усвоил знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с использованием ОоП в IDE PyCharm Community.