Студент группы ИС-25 Неговора Н.М.

**Практическое занятие № 16**

**Тема:** составление программ с использованием ООП.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с ООП в IDE PyCharm Community

**Постановка задачи:**

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

2. 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".

3. Для задачи из блока 1 создать две функции, save\_def и load\_def, которые позволяют сохранять информацию из экземпляров класса в файл и загружать ее обратно. Использовать модуль pickle для сериализации и десериализации объектов Python в бинарном формате.

**Тип Алгоритма**: Линейный

**Текст программы:**

import calendar

import re

class CALENDAR():

def \_\_init\_\_(self, day = 15, month = 10, year = 2006) -> None:

self.year = year

self.month = month

self.day = day

self.day\_name = {

0: "Понедельник",

1: "Вторник",

2: "Среда",

3: "Четверг",

4: "Пятница",

5: "Суббота",

6: "Воскресенье"

}

def get\_day\_of\_week(self):

"""

Возвращает день недели на русском языке

"""

day = calendar.weekday(self.year, self.month, self.day)

day = self.day\_name[day]

return day

def check\_for\_leap\_year(self):

"""

Возвращает строку "Весокосный" или "Не весокосный", по отношению к году, указаному в классе

"""

result = calendar.isleap(self.year)

if result: return "Весокосный"

else: return "Не весокосный"

def get\_umber\_of\_days\_in\_a\_month(self):

"""

Возвращает строку "Дней в месяце: {data}", где date - количество дней в месяце

"""

data = calendar.monthrange(self.year, self.month)[1]

return f"Дней в месяце: {data}"

if \_\_name\_\_ == "\_\_main\_\_":

cal = CALENDAR()

print(cal.get\_day\_of\_week())

print(cal.check\_for\_leap\_year())

print(cal.get\_umber\_of\_days\_in\_a\_month())

**Протокол работы программы 1:**

from typing import Any

class ANIMAL():

def \_\_init\_\_(self, kind = "Животное", number\_of\_paws = 4, color = "Белый"):

self.kind = kind

self.number\_of\_paws = number\_of\_paws

self.color = color

class DOG(ANIMAL):

def \_\_init\_\_(self, kind="Собака", number\_of\_paws=4, color="Белый", name="Боб", breed="Дворняга"):

super().\_\_init\_\_(kind, number\_of\_paws, color)

self.name = name

self.breed = breed

def get\_all\_attribute(self) -> list[str]:

list\_attr = [attr for attr in self.\_\_dict\_\_ if not callable(getattr(self, attr)) and not attr.startswith("\_\_")]

list\_attr = [f" {item}: {self.\_\_getattribute\_\_(item)}" for item in list\_attr]

return list\_attr

if \_\_name\_\_ == "\_\_main\_\_":

bob = DOG(name="Боб", breed="Немецкая Гончая")

bob\_description = bob.get\_all\_attribute()

print("\nОписание Боба:")

print(\*bob\_description, sep="\n")

print("\n")

**Протокол работы программы 2:**

Russian\_String = Строка состоящая из русских слов

Replace Russian\_String = Строка.состоящая.из.русских.слов"

Process finished with exit code 0

import pickle

from task\_1 import CALENDAR as OLD\_CALENDAR

from task\_2 import DOG as OLD\_DOG

class CALENDAR(OLD\_CALENDAR):

def \_\_init\_\_(self, day=15, month=10, year=2006):

super().\_\_init\_\_(day, month, year)

class DOG(OLD\_DOG):

def \_\_init\_\_(self, kind="Собака", number\_of\_paws=4, color="Белый", name="Боб", breed="Дворняга"):

super().\_\_init\_\_(kind, number\_of\_paws, color, name, breed)

def save\_def(obj: object):

with open(f"{obj.\_\_class\_\_.\_\_name\_\_}.pickle", "wb") as file:

pickle.dump(obj, file)

def load\_def(class\_name):

with open(f"{class\_name}.pickle", "rb") as file:

result = pickle.load(file)

return result

if \_\_name\_\_ == "\_\_main\_\_":

save\_def(DOG(name="Игорь"))

igor = load\_def("DOG")

igor\_description = igor.get\_all\_attribute()

print(\*igor\_description, sep="\n")

**Вывод**: В данной практической работе Я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составление программ с функциями в IDE PyCharm Community.

.