

Практическая работа №16

Тема: Составление программ с использованием ООП.

Цель: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления 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.

Тип алгоритма: Линейный

Текст программы:

```
import datetime
import calendar

class Calendar:
    def __init__(self, day: int, month: int, year: int) -> None:
        self.year = year
        self.month = month
        self.day = day
        try:
            self.date = datetime.datetime(self.year, self.month, self.day)
        except ValueError:
            print("day is out of range for month\n")

    def day_of_week(self) -> str:
        days_of_week = {
            1: "Понедельник",
            2: "Вторник",
            3: "Среда",
            4: "Четверг",
            5: "Пятница",
            6: "Суббота",
            7: "Воскресенье",
        }
        try:
            return days_of_week[self.date.isoweekday()]
        except AttributeError:
            return "Неверный формат даты"

    def is_leap_year(self) -> str:
        return "Високосный" if self.year % 4 == 0 else "Не високосный"

    def days_in_month(self) -> int:
        return calendar.monthrange(self.year, self.month)[1]

date = Calendar(27, 2, 2024)
print(
    f"День недели: {date.day_of_week()}\n"
    f"Год: {date.is_leap_year()}\n"
    f"Количество дней в месяце: {date.days_in_month()}"
)
```

Протокол работы программы:

День недели: Вторник

Год: Високосный

Количество дней в месяце: 29

Process finished with exit code 0

Постановка задачи №1.

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".

Тип алгоритма: Линейный

Текст программы:

```
class Animal:
    def __init__(self, vid: str, amount_of_paws: int, cool_of_wool: str):
        self.vid = vid
        self.amount_of_paws = amount_of_paws
        self.cool_of_wool = cool_of_wool

    def __str__(self):
        return f"Вид: {self.vid},\nЛап: {self.amount_of_paws}шт,\nШерсть: {self.cool_of_wool}"

class Dog(Animal):
    def __init__(self, vid: str, amount_of_paws: int, cool_of_wool: str,
nickname: str, breed: str):
        super().__init__(vid, amount_of_paws, cool_of_wool)
        self.nickname = nickname
        self.breed = breed

    def __str__(self):
        return f"\n{super().__str__()},\nКличка: {self.nickname},\nПорода: {self.breed}"

animal = Animal(vid="Животное", amount_of_paws=0, cool_of_wool="😊🌀")
print(animal)
dog = Dog(vid="Собака", amount_of_paws=4, cool_of_wool="Белая",
nickname="Собака", breed="Собака")
print(dog)
```

Протокол работы программы:

Вид: Животное,

Лап: 0шт,

Шерсть: 😊🌀

Вид: Собака,

Лап: 4шт,

Шерсть: Белая,

Кличка: Собака,

Порода: Собака

Process finished with exit code 0

Вывод: В процессе выполнения практического занятия №16 закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с ООП в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные кода выложены на GitHub.

