

Практическое занятие № 17

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

Постановка задачи. Создайте класс "Машина" с атрибутами "марка", "модель" и "год выпуска". Напишите метод, который выводит информацию о машине в формате "Марка: марка, Модель: модель, Год выпуска: год"

Тип алгоритма:циклический

Текст программы:

```
#Создайте класс "Машина" с атрибутами "марка", "модель" и "год выпуска". Напишите метод, который выводит и
import tkinter as tk

class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year

    def display_info(self):
        return f"Марка: {self.make}, Модель: {self.model}, Год выпуска: {self.year}"

def show_car_info():
    make = make_entry.get()
    model = model_entry.get()
    year = year_entry.get()
    car = Car(make, model, year)
    result_label.config(text=car.display_info())

# Создание главного окна
root = tk.Tk()
root.title("Car Information")

# Ввод данных
tk.Label(root, text="Марка").grid(row=0)
tk.Label(root, text="Модель").grid(row=1)
tk.Label(root, text="Год выпуска").grid(row=2)

make_entry = tk.Entry(root)
model_entry = tk.Entry(root)
year_entry = tk.Entry(root)

make_entry.grid(row=0, column=1)
model_entry.grid(row=1, column=1)
year_entry.grid(row=2, column=1)

# Кнопка для отображения информации
tk.Button(root, text="Показать информацию", command=show_car_info).grid(row=3, column=1)

# Метка для вывода результата
result_label = tk.Label(root, text="")
result_label.grid(row=4, columnspan=2)

# Запуск основного цикла приложения
root.mainloop()
```

```

#Создайте базовый класс "Форма" со свойствами "цвет" и "тип". От этого класса унаследуйте класс "Круг" и доба
import tkinter as tk
import math

class Shape:
    def __init__(self, color, shape_type):
        self.color = color
        self.shape_type = shape_type

    def display_info(self):
        return f"Цвет: {self.color}, Тип: {self.shape_type}"

class Circle(Shape):
    def __init__(self, color, radius):
        super().__init__(color, "Круг")
        self.radius = radius

    def area(self):
        return math.pi * (self.radius ** 2)

    def perimeter(self):
        return 2 * math.pi * self.radius

    def display_info(self):
        return (super().display_info() +
                f", Радиус: {self.radius}, Площадь: {self.area():.2f}, Периметр: {self.perimeter():.2f}")

def show_circle_info():
    color = color_entry.get()
    radius = float(radius_entry.get())
    circle = Circle(color, radius)
    result_label.config(text=circle.display_info())

# Создание главного окна
root = tk.Tk()
root.title("Circle Information")

# Ввод данных
tk.Label(root, text="Цвет").grid(row=0)
tk.Label(root, text="Радиус").grid(row=1)

color_entry = tk.Entry(root)
radius_entry = tk.Entry(root)

color_entry.grid(row=0, column=1)
radius_entry.grid(row=1, column=1)

# Кнопка для отображения информации
tk.Button(root, text="Показать информацию", command=show_circle_info).grid(row=2, column=1)

# Метка для вывода результата
result_label = tk.Label(root, text="")
result_label.grid(row=3, columnspan=2)

# Запуск основного цикла приложения
root.mainloop()

```

Протокол работы программы:

Process finished with exit code 0

Вывод: В процессе выполнения закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с ООП в IDE PyCharm Community. Были использованы языковые конструкции `import`, `def`, `return`

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на [GitHub](#)