

Практическое задание №17.1

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: Создайте класс "Компьютер" с атрибутами "марка", "процессор" и "оперативная память". Напишите метод, который выводит информацию о компьютере в формате "Марка: марка, Процессор: процессор, Оперативная память: память".

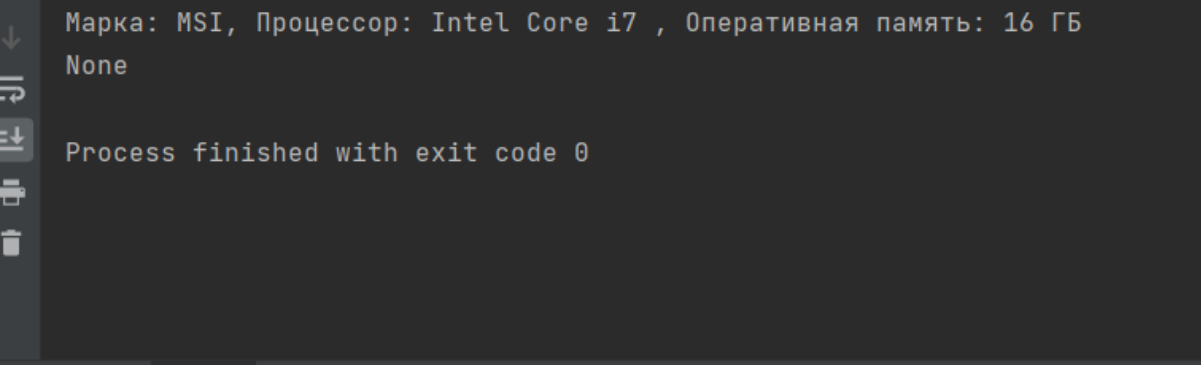
Тест программы

```
class Komputer:
    def __init__(self, stamp, processor, ram):
        self.__stamp = stamp
        self.__processor = processor
        self.__ram = ram

    def get_info(self):
        return print(f"Марка: {self.__stamp},
Процессор: {self.__processor} , Оперативная память:
{self.__ram}")

my_komp = Komputer("MSI", "Intel Core i7", "16 ГБ")
print(my_komp.get_info())
```

Протокол работы программы:

A screenshot of a terminal window with a dark background. The output text is as follows:

```
Марка: MSI, Процессор: Intel Core i7 , Оперативная память: 16 ГБ  
None  
  
Process finished with exit code 0
```

Практическое задание №17.2

Постановка задачи: Создайте базовый класс "Человек" со свойствами "имя", "возраст" и "пол". От этого класса унаследуйте классы "Мужчина" и "Женщина" и добавьте в них свойства, связанные с социальным положением (например, "семейное положение", "количество детей" и т.д.).

Тест программы

```
# Создайте базовый класс "Человек" со свойствами  
"имя", "возраст" и "пол". От этого  
# класса унаследуйте классы "Мужчина" и "Женщина" и  
добавьте в них свойства,  
# связанные с социальным положением (например,  
"семейное положение",  
# "количество детей" и т.д.).
```

```
class Human:
```

```
def __init__(self, name, age, gender):
    self.name = name
    self.age = age
    self.gender = gender

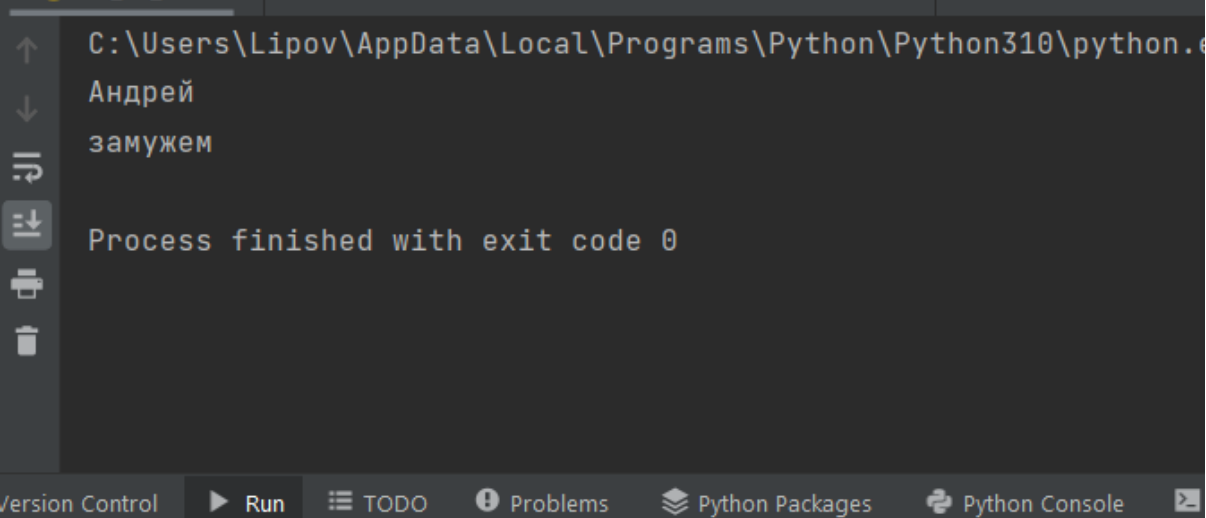
class Man(Human):
    def __init__(self, name, age, family_status,
number_of_children):
        super().__init__(name, age, "male")
        self.family_status = family_status
        self.number_of_children = number_of_children

class Woman(Human):
    def __init__(self, name, age, family_status,
number_of_children):
        super().__init__(name, age, "female")
        self.family_status = family_status
        self.number_of_children = number_of_children

man1 = Man("Андрей", 28, "женат", 3)
woman1 = Woman("Катя", 22, "замужем", 1)

print(man1.name)    # Андрей
print(woman1.family_status)  # замужем
```

Протокол работы программы:



The screenshot shows a Python console window with a dark background. On the left, there is a vertical toolbar with icons for navigation (up, down), search, and other functions. The main area of the console displays the following text:

```
C:\Users\Lipov\AppData\Local\Programs\Python\Python310\python.exe
Андрей
замужем

Process finished with exit code 0
```

At the bottom of the window, there is a status bar with several tabs: "Version Control", "Run", "TODO", "Problems", "Python Packages", "Python Console", and a mail icon.