Студент группы ИС-25 Рычков Д.В.

**Практическое задание 16-27**

**1. Наименование практического занятия:** составление программ с использованием ООП.

**2. Количество часов:** 4

**3. Место проведения:** главный корпус РКСИ, ауд. 420.

**4. Цели практического занятия:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием списковых включений, итераторов, генераторов в IDE PyCharm Community.

***Текст программы:***

***#Вариант 27***.

Задача №1

*"""27. Создайте класс «Студент», который имеет атрибуты имя, фамилия и оценки.*  
*Добавьте методы для вычисления среднего балла и определения, является ли студент*  
*отличником."""*  
  
  
  
class Student:  
 def \_\_init\_\_(self, first\_name, last\_name, grades):  
 self.first\_name = first\_name  
 self.last\_name = last\_name  
 self.grades = grades  
  
 def calculate\_average\_grade(self):  
 if len(self.grades) == 0:  
 return 0  
 return sum(self.grades) / len(self.grades)  
  
 def is\_excellent\_student(self):  
 average\_grade = self.calculate\_average\_grade()  
 return average\_grade >= 5  
  
  
student1 = Student("Иван", "Иванов", [5, 5, 5, 5, 5])  
student2 = Student("Петр", "Петров", [3, 4, 5, 4, 3])  
  
print(f"{student1.first\_name} {student1.last\_name} средний балл: {student1.calculate\_average\_grade()}")  
print(f"{student1.first\_name} {student1.last\_name} является отличником: {student1.is\_excellent\_student()}")  
  
print(f"{student2.first\_name} {student2.last\_name} средний балл: {student2.calculate\_average\_grade()}")  
print(f"{student2.first\_name} {student2.last\_name} является отличником: {student2.is\_excellent\_student()}")

***Протокол работы программы:***

Иван Иванов средний балл: 5.0

Иван Иванов является отличником: True

Петр Петров средний балл: 3.8

Петр Петров является отличником: False

*Задача №2*

*"""Создайте класс "Автомобиль", который содержит информацию о марке, модели и  
годе выпуска. Создайте класс "Грузовик", который наследуется от класса  
"Автомобиль" и содержит информацию о грузоподъемности. Создайте класс  
"Легковой автомобиль", который наследуется от класса "Автомобиль" и содержит  
информацию о количестве пассажиров."""*class Car: def \_\_init\_\_(self, brand, model, year): self.brand = brand self.model = model self.year = yearclass Truck(Car): def \_\_init\_\_(self, brand, model, year, capacity): super().\_\_init\_\_(brand, model, year) self.capacity = capacityclass PassengerCar(Car): def \_\_init\_\_(self, brand, model, year, passengers): super().\_\_init\_\_(brand, model, year) self.passengers = passengerstruck1 = Truck('Volvo', 'VNL', 2021, 20000)print(truck1.brand, truck1.model, truck1.year, truck1.capacity)car1 = PassengerCar('Toyota', 'Camry', 2019, 5)print(car1.brand, car1.model, car1.year, car1.passengers)

***Протокол работы программы:***

*/usr/bin/python3.9 /home/student/Рабочий стол/PZ-16/PZ-16-2.py*

*Volvo VNL 2021 20000*

*Toyota Camry 2019 5*

*Process finished with exit code 0*

*Задача №3*

import pickleclass Student: def \_\_init\_\_(self, first\_name, last\_name, grades): self.first\_name = first\_name self.last\_name = last\_name self.grades = grades def calculate\_average\_grade(self): if len(self.grades) == 0: return 0 return sum(self.grades) / len(self.grades) def is\_excellent\_student(self): average\_grade = self.calculate\_average\_grade() return average\_grade >= 5def save\_def(students): with open('students\_info.pickle', 'wb') as file: pickle.dump(students, file)def load\_def(): with open('students\_info.pickle', 'rb') as file: return pickle.load(file)# Создание объектов класса Studentstudent1 = Student("Иван", "Иванов", [5, 4, 4, 5, 5])student2 = Student("Петр", "Петров", [4, 3, 5, 4, 4])student3 = Student("Мария", "Сидорова", [5, 5, 5, 5, 5])# Сохранение информации об объектах в файлsave\_def([student1, student2, student3])# Загрузка информации из файлаloaded\_students = load\_def()for student in loaded\_students: print(f"{student.first\_name} {student.last\_name}:") print(f"Оценки: {student.grades}") print(f"Средний балл: {student.calculate\_average\_grade()}") if student.is\_excellent\_student(): print("Студент отличник") else: print("Студент не отличник") print()

***Протокол работы программы:***

*/usr/bin/python3.9 /home/student/Рабочий стол/PZ-16/PZ-16-3.py*

*Иван Иванов:*

*Оценки: [5, 4, 4, 5, 5]*

*Средний балл: 4.6*

*Студент не отличник*

*Петр Петров:*

*Оценки: [4, 3, 5, 4, 4]*

*Средний балл: 4.0*

*Студент не отличник*

*Мария Сидорова:*

*Оценки: [5, 5, 5, 5, 5]*

*Средний балл: 5.0*

*Студент отличник*

***Вывод:*** закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составление программ с использованием списковых включений, итераторов, генераторов в IDE PyCharm Community.