Варіант№1 Розробити систему для контролю звітності по заробітній платі для працівників команди фірми-розробника ПЗ. Усі працівники поділяються на штатних і найманих підрядників. Кожна з двох груп має свою систему нарахування заробітної плати і визначення кількості робочих годин в тижнем. Не залежно від групи працівники можуть мати певний рівень кваліфікації (Trainee , Junior, Middle, Senior). Система нарахувань заробітної плати досить гнучка, а керівник команди може визначати ще якийсь проміжних рівень кваліфікації (це залежить від настрою на момент присудження кваліфікації). Також існує система бонусів, якою можуть скористатись всі працівники в якості винагороди. Розробити програму, яка б дозволяла змінювати рівень кваліфікації працівника та відповідно до цього нараховувати йому заробітну плату, формувати список десяти найкращих працівників періоду за кількістю відпрацьованих понаднормових годин та виділяти їм премію у розмірі окладу. Скористатись шаблонами Фасад та Декоратор.

from abc import ABC, abstractmethod

# Декоратор для бонусів

class BonusDecorator(ABC):

    def \_\_init\_\_(self, employee):

        self.employee = employee

    def calculate\_salary(self):

        return self.employee.calculate\_salary()

    def \_\_str\_\_(self):

        return f"{self.employee.name} ({self.employee.qualification}) - Salary with bonus: {self.calculate\_salary()}"

# Основний клас працівника

class Employee(ABC):

    def \_\_init\_\_(self, name, qualification, base\_salary, hours\_per\_week):

        self.name = name

        self.qualification = qualification

        self.base\_salary = base\_salary

        self.hours\_per\_week = hours\_per\_week

        self.overtime\_hours = 0

    @abstractmethod

    def calculate\_salary(self):

        pass

    def add\_overtime(self, hours):

        self.overtime\_hours += hours

    def \_\_str\_\_(self):

        return f"{self.name} ({self.qualification}) - Salary: {self.calculate\_salary()}"

# Штатні працівники

class FullTimeEmployee(Employee):

    def calculate\_salary(self):

        return self.base\_salary + (self.overtime\_hours \* 20)  # Доплата за понаднормові

# Підрядники

class Contractor(Employee):

    def calculate\_salary(self):

        return self.base\_salary + (self.overtime\_hours \* 30)  # Інша доплата за понаднормові

# Декоратори бонусів

class Bonus(BonusDecorator):

    def calculate\_salary(self):

        return self.employee.calculate\_salary() \* 1.1  # 10% бонус

# Фасад для керування працівниками

class PayrollSystem:

    def \_\_init\_\_(self):

        self.employees = [

            FullTimeEmployee("Олексій", "Senior", 5000, 40),

            FullTimeEmployee("Марина", "Middle", 4000, 40),

            FullTimeEmployee("Ігор", "Junior", 3000, 40),

            Contractor("Анна", "Middle", 3500, 30),

            Contractor("Сергій", "Senior", 6000, 35),

            FullTimeEmployee("Дмитро", "Trainee", 2500, 40),

            Contractor("Ольга", "Junior", 3200, 25),

            FullTimeEmployee("Євген", "Middle", 4200, 40),

            FullTimeEmployee("Наталія", "Senior", 5500, 40),

            Contractor("Віктор", "Middle", 3800, 30)

        ]

    def add\_employee(self, employee):

        self.employees.append(employee)

    def change\_qualification(self, employee\_name, new\_qualification):

        for emp in self.employees:

            if emp.name == employee\_name:

                emp.qualification = new\_qualification

                break

    def top\_overtime\_employees(self):

        top\_10 = sorted(self.employees, key=lambda e: e.overtime\_hours, reverse=True)[:10]

        return [Bonus(emp) for emp in top\_10]

    def print\_salaries(self):

        for emp in self.employees:

            print(emp)

    def print\_all\_employees(self):

        for emp in self.employees:

            print(emp.name, "-", emp.qualification)

def main():

    payroll = PayrollSystem()

    while True:

        print("\nМеню:")

        print("1. Додати працівника")

        print("2. Змінити рівень кваліфікації працівника")

        print("3. Додати понаднормові години")

        print("4. Обчислити зарплату всіх працівників")

        print("5. Визначити 10 найкращих працівників за понаднормовими годинами")

        print("6. Переглянути всіх працівників")

        print("7. Вийти")

        choice = input("Оберіть опцію: ")

        if choice == "1":

            name = input("Введіть ім'я працівника: ")

            qualification = input("Введіть рівень кваліфікації: ")

            base\_salary = int(input("Введіть базову зарплату: "))

            hours\_per\_week = int(input("Введіть кількість годин на тиждень: "))

            emp\_type = input("Тип працівника (fulltime/contractor): ").strip().lower()

            if emp\_type == "fulltime":

                employee = FullTimeEmployee(name, qualification, base\_salary, hours\_per\_week)

            elif emp\_type == "contractor":

                employee = Contractor(name, qualification, base\_salary, hours\_per\_week)

            else:

                print("Невідомий тип працівника!")

                continue

            payroll.add\_employee(employee)

            print("Працівника додано!")

        elif choice == "2":

            name = input("Введіть ім'я працівника: ")

            new\_qualification = input("Введіть новий рівень кваліфікації: ")

            payroll.change\_qualification(name, new\_qualification)

            print("Кваліфікацію оновлено!")

        elif choice == "3":

            name = input("Введіть ім'я працівника: ")

            hours = int(input("Введіть кількість понаднормових годин: "))

            for emp in payroll.employees:

                if emp.name == name:

                    emp.add\_overtime(hours)

                    print("Понаднормові години додано!")

                    break

            else:

                print("Працівник не знайдений!")

        elif choice == "4":

            payroll.print\_salaries()

        elif choice == "5":

            print("Топ 10 працівників за понаднормовими годинами та їх премія:")

            for emp in payroll.top\_overtime\_employees():

                print(emp)

        elif choice == "6":

            print("Список всіх працівників:")

            payroll.print\_all\_employees()

        elif choice == "7":

            print("Вихід...")

            break

        else:

            print("Невідома опція, спробуйте ще раз!")

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

    main()

