

```

class myDate{
    //приватные поля
    #nameDaysOfWeek =
['понедельник', 'вторник', 'среда', 'четверг', 'пятница', 'суббота', 'воскресенье'];
    #nameMonth =
['января', 'февраля', 'марта', 'апреля', 'мая', 'июня', 'июля', 'августа', 'сентября', 'ок
тября', 'ноября', 'декабря']
    constructor(day, month, year){
        this.day = day;
        this.month = month;
        this.year = year;
    }
    //проверяем является ли год високосным
    IsLeapYear(year){
        var a = ((year%4===0 && year%100!==0) || year%400===0);
        return ((year%4===0 && year%100!==0) || year%400===0);
    }
    //получаем кол-во
    GetCountDayOfMonth(month, year){
        if (month==2){
            return (this.IsLeapYear(year)) ? 29:28;
        }
        return month<8 ? (30 + month%2) : (30 + (month+1)%2)
    }
    //получаем количество дней до текущей даты в году включая текущую
    GetCountYearDayBefo(){
        if(this.month==1){
            return this.day
        }
        var result = 0;
        for(let i=1; i<this.month; i++){
            result+=this.GetCountDayOfMonth(i, this.year);
        }
        return result+this.day;
    }
    //получение количества дней до конца года без текущей
    GetCountYearDayAfte(){
        if(this.month==12){
            return 31-this.day;
        }
        var result = this.GetCountDayOfMonth(this.month, this.year)-this.day;
        for(let i = this.month+1; i<=12; i++){
            result+=this.GetCountDayOfMonth(i, this.year);
        }
        return result;
    }
    //получение кол-во дней года
    GetYearDayCount(year){
        return this.IsLeapYear(year) ? 366 : 365;
    }
    //Сравнение дат -1 otherDay меньше 0 равны 1 больше

```

```

MatchDate(otheDate){
    if(otheDate.year<this.year)
        return -1;
    if(otheDate.year>this.year)
        return 1;
    if(otheDate.month<this.month)
        return -1;
    if(otheDate.month>this.month)
        return 1;
    if(otheDate.day<this.day)
        return -1;
    if(otheDate.day>this.day)
        return 1;
    return 0;
}
//Получение количества дней между дтатами
GetCountDayBetwin(otheDay){
    if( this.MatchDate(otheDay)==0)
        return 0;
    var startDate = this.MatchDate(otheDay)<0? otheDay: this;
    var endDate = this.MatchDate(otheDay)>0? otheDay: this;
    var result =
startDate.GetCountYearDayAfte()+endDate.GetCountYearDayBefo();

    for(let i = startDate.year+1; i<endDate.year; i++){
        result += this.GetYearDayCount(i);
    }
    return result;
}
//получение дня недели
GetWeekDay(){
    var baseDate= new myDate(3,1,2000);

    return this.#nameDaysOfWeek[(this.GetCountDayBetwin(baseDate))%7]
}
//дата в строке
GetStringDate(){
    return `${this.GetWeekDay()} ${this.day} ${this.#nameMoth[this.month-1]}
${this.year}`;
}
}

const curentDay = new myDate(5,12,2024);
const otheDate = new myDate(26,8,1996);
alert(`${otheDate.GetStringDate()} - ${curentDay.GetStringDate()} разница =
${curentDay.GetCountDayBetwin(otheDate)}` );

```