```
const std = @import("std");

const dds = @import("dds");

// keyboard
const kbd = @import("cursed").kbd;

// panel
const pnl = @import("forms").pnl;
// button
const btn = @import("forms").btn;
// label
const lbl = @import("forms").lbl;
// menu
const mnu = @import("forms").mnu;
// flied
const fld = @import("forms").fld;
// line horizontal
const lnh = @import("forms").lnh;
// line vertival
const lnv = @import("forms").lnv;

// grid
const grd = @import("grid").grd;

// full delete for produc
const forms = @import("forms");
const allocator = std.heap.page_allocator;

pub fn SavJson(XPANEL: std.ArrayList(pnl.PANEL)) !void {
    var out_buf: [20480]u8 = undefined;
    var slice_stream = std.io.fixedBufferStream(&out_buf);
    const out = slice_stream.writer();

    var w = std.json.writeStream(out, .{ .whitespace = .indent_2 });

    const Ipanel = std.enums.EnumIndexer(pnl.Epanel);
    try w.beginObject();
    try w.objectField("PANEL");
    var nbrPnl: usize = XPANEL.items.len;
    var np: usize = 0;
    while (np < nbrPnl) : (np += 1) {
        try w.beginArray();
        try w.beginObject();
        var p: usize = 0;
        while (p < Ipanel.count) : (p += 1) {
            switch (Ipanel.keyForIndex(p)) {
                .name => {
                    try w.objectField(@tagName(pnl.Epanel.name));
                    try w.print("\"{s}\"", .{XPANEL.items[np].name});
                },
                .posx => {
                    try w.objectField(@tagName(pnl.Epanel.posx));
                    try w.print("{d}", .{XPANEL.items[np].posx});
                },
                .posy => {
                    try w.objectField(@tagName(pnl.Epanel.posy));
                    try w.print("{d}", .{XPANEL.items[np].posy});
                },
                .lines => {
                    try w.objectField(@tagName(pnl.Epanel.lines));
                    try w.print("{d}", .{XPANEL.items[np].lines});
                },
                .cadre => {
                    try w.objectField(@tagName(pnl.Epanel.cadre));
                    try w.print("{s}", .{@tagName(XPANEL.items[np].frame.cadre)});
                },
                .title => {
                    try w.objectField(@tagName(pnl.Epanel.title));
                    try w.print("{s}", .{XPANEL.items[np].frame.title});
                },
                .button => {
                    const Ibutton = std.enums.EnumIndexer(btn.Ebutton);
                    var nbrBtn: usize = XPANEL.items[np].button.items.len;
                    var bp: usize = 0;
                    try w.objectField("button");
                    try w.beginArray();
                    while (bp < nbrBtn) : (bp += 1) {
                        try w.beginObject();
                        var b: usize = 0;
                        while (b < Ibutton.count) : (b += 1) {
                            switch (Ibutton.keyForIndex(b)) {
                                .name => {
                                    try w.objectField(@tagName(btn.Ebutton.name));
                                    try w.print("\"{s}\"", .{XPANEL.items[np].button.items[bp].name});
                                },
                                .key => {
                                    try w.objectField(@tagName(btn.Ebutton.key));
                                    try w.print("\"{s}\"", .{@tagName(XPANEL.items[np].button.items[bp].key)});
                                },
                                .show => {
                                    try w.objectField(@tagName(btn.Ebutton.show));
                                    try w.print("{d}", .{@intFromBool(XPANEL.items[np].button.items[bp].show)});
                                },
                                .check => {
                                    try w.objectField(@tagName(btn.Ebutton.check));
                                    try w.print("{d}", .{@intFromBool(XPANEL.items[np].button.items[bp].check)});
                                },
                                .title => {
                                    try w.objectField(@tagName(btn.Ebutton.title));
                                    try w.print("\"{s}\"", .{XPANEL.items[np].button.items[bp].title});
                                },
                            }
                        }
                        try w.endObject();
                    }

                    try w.endArray();
                },
                .label => {
                    const Ilabel = std.enums.EnumIndexer(lbl.Elabel);
                    var l: usize = 0;
                    var nbrLbl: usize = XPANEL.items[np].label.items.len;

                    var lp: usize = 0;
                    try w.objectField("label");
                    try w.beginArray();
                    while (lp < nbrLbl) : (lp += 1) {
                        try w.beginObject();
                        l = 0;
                        while (l < Ilabel.count) : (l += 1) {
                            switch (Ilabel.keyForIndex(l)) {
                                .name => {
                                    try w.objectField(@tagName(lbl.Elabel.name));
                                    try w.print("\"{s}\"", .{XPANEL.items[np].label.items[lp].name});
                                },
                                .posx => {
                                    try w.objectField(@tagName(lbl.Elabel.posx));
                                    try w.print("{d}", .{XPANEL.items[np].label.items[lp].posx});
                                },
                                .posy => {
                                    try w.objectField(@tagName(lbl.Elabel.posy));
                                    try w.print("{d}", .{XPANEL.items[np].label.items[lp].posy});
                                },
                                .text => {
                                    try w.objectField(@tagName(lbl.Elabel.title));
                                    try w.print("\"{s}\"", .{XPANEL.items[np].label.items[lp].text});
                                },
                                .title => {
                                    try w.objectField(@tagName(lbl.Elabel.title));
                                    try w.print("{d}", .{@intFromBool(XPANEL.items[np].label.items[lp].title)});
                                },
                            }
                        }
                        try w.endObject();
                    }

                    try w.endArray();
                },
                .field => {
                    const Ifield = std.enums.EnumIndexer(fld.Efield);
                    var f: usize = 0;
                    var nbrFld: usize = XPANEL.items[np].field.items.len;

                    var fp: usize = 0;
                    try w.objectField("field");
                    try w.beginArray();
                    while (fp < nbrFld) : (fp += 1) {
                        try w.beginObject();
                        f = 0;
                        while (f < Ifield.count) : (f += 1) {
                            switch (Ifield.keyForIndex(f)) {
                                .name => {
                                    try w.objectField(@tagName(fld.Efield.name));
                                    try w.print("\"{s}\"", .{XPANEL.items[np].field.items[fp].name});
                                },
                                .posx => {
                                    try w.objectField(@tagName(fld.Efield.posx));
                                    try w.print("{d}", .{XPANEL.items[np].field.items[fp].posx});
                                },
                                .posy => {
                                    try w.objectField(@tagName(fld.Efield.posy));
                                    try w.print("{d}", .{XPANEL.items[np].field.items[fp].posy});
                                },
                                .reftyp => {
                                    try w.objectField(@tagName(fld.Efield.reftyp));
                                    try w.print("{s}", .{@tagName(XPANEL.items[np].field.items[fp].reftyp)});
                                },
                                .width => {
                                    try w.objectField(@tagName(fld.Efield.width));
                                    try w.print("{d}", .{XPANEL.items[np].field.items[fp].width});
                                },
                                .scal => {
                                    try w.objectField(@tagName(fld.Efield.scal));
```

```
                                            try w.print("{d}", .{XPANEL.items[np].field.items[fp].scal});
                                        },
                                        .requier => {
                                            try w.objectField(@tagName(fld.Efield.requier));
                                            try w.print("{d}", .{@intFromBool(XPANEL.items[np].field.items[fp].requier)});
                                        },
                                        .protect => {
                                            try w.objectField(@tagName(fld.Efield.protect));
                                            try w.print("{d}", .{@intFromBool(XPANEL.items[np].field.items[fp].protect)});
                                        },
                                        .edtcar => {
                                            try w.objectField(@tagName(fld.Efield.edtcar));
                                            try w.print("\"{s}\"", .{XPANEL.items[np].field.items[fp].edtcar});
                                        },
                                        .errmsg => {
                                            try w.objectField(@tagName(fld.Efield.errmsg));
                                            try w.print("\"{s}\"", .{XPANEL.items[np].field.items[fp].errmsg});
                                        },
                                        .help => {
                                            try w.objectField(@tagName(fld.Efield.help));
                                            try w.print("\"{s}\"", .{XPANEL.items[np].field.items[fp].help});
                                        },
                                        .procfunc => {
                                            try w.objectField(@tagName(fld.Efield.procfunc));
                                            try w.print("\"{s}\"", .{XPANEL.items[np].field.items[fp].procfunc});
                                        },
                                        .proctask => {
                                            try w.objectField(@tagName(fld.Efield.proctask));
                                            try w.print("\"{s}\"", .{XPANEL.items[np].field.items[fp].proctask});
                                        },
                                    }
                                }
                                try w.endObject();
                            }

                            try w.endArray();
                        },
                        else => {},
                    }
                }
                try w.endObject();
                try w.endArray();
            }
            try w.endObject();

            const result = slice_stream.getWritten();
```

```

    var my_file = try std.fs.cwd().createFile("Zdspf.txt", .{ .read = true });
    _ = try my_file.write(result);
    my_file.close();

    _ = kbd.getKEY();
    slice_stream.reset();
}
```