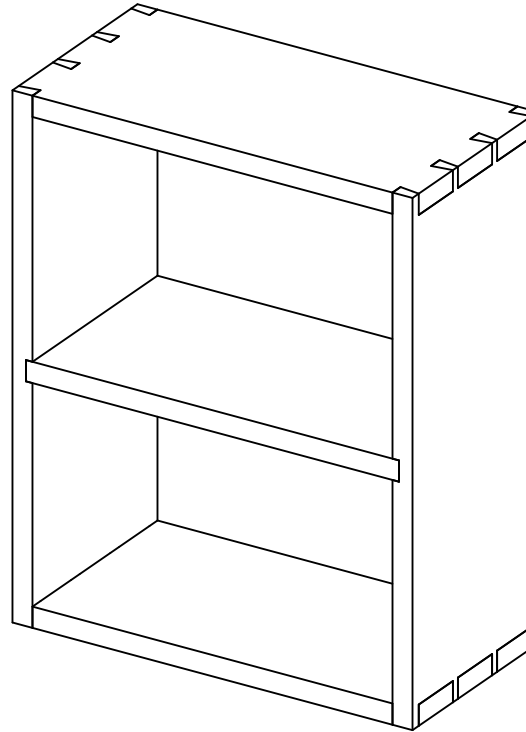


Dovetail shelf



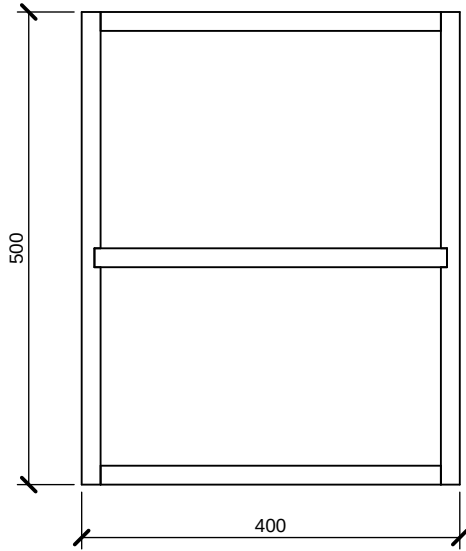
parts	5 (3 different)
overall	400 x 200 x 500 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

Contents

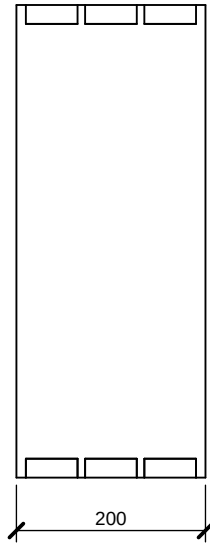
Drawings	3
Assembly	3
Parts	4
1 left_right	5
2 top_bottom	6
3 shelf	7
 Cut list	 8
 Appendix: source	 9



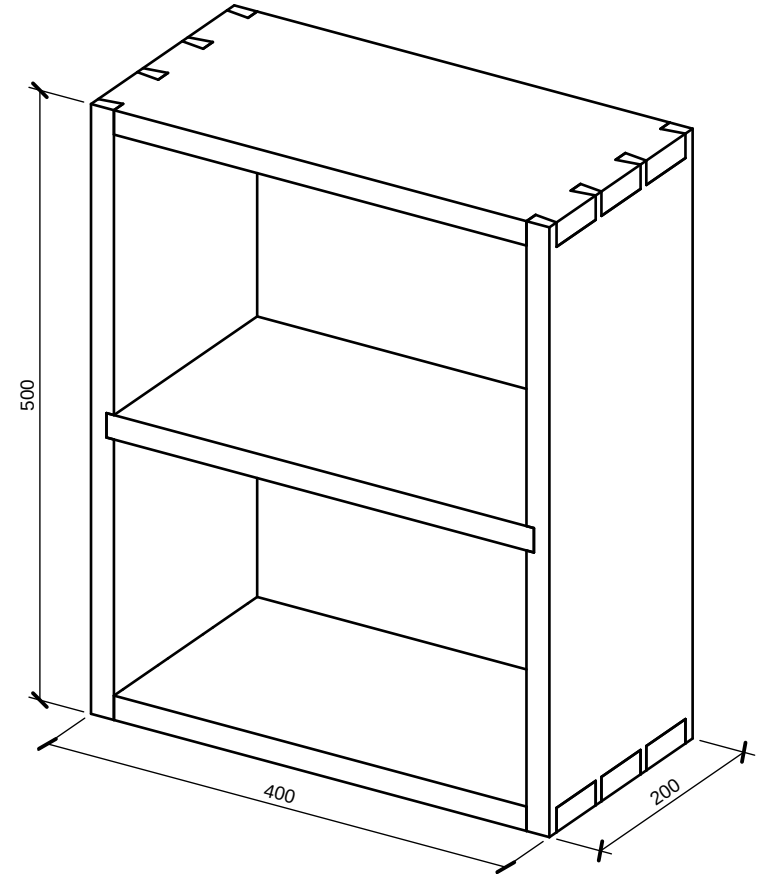
TOP



FRONT

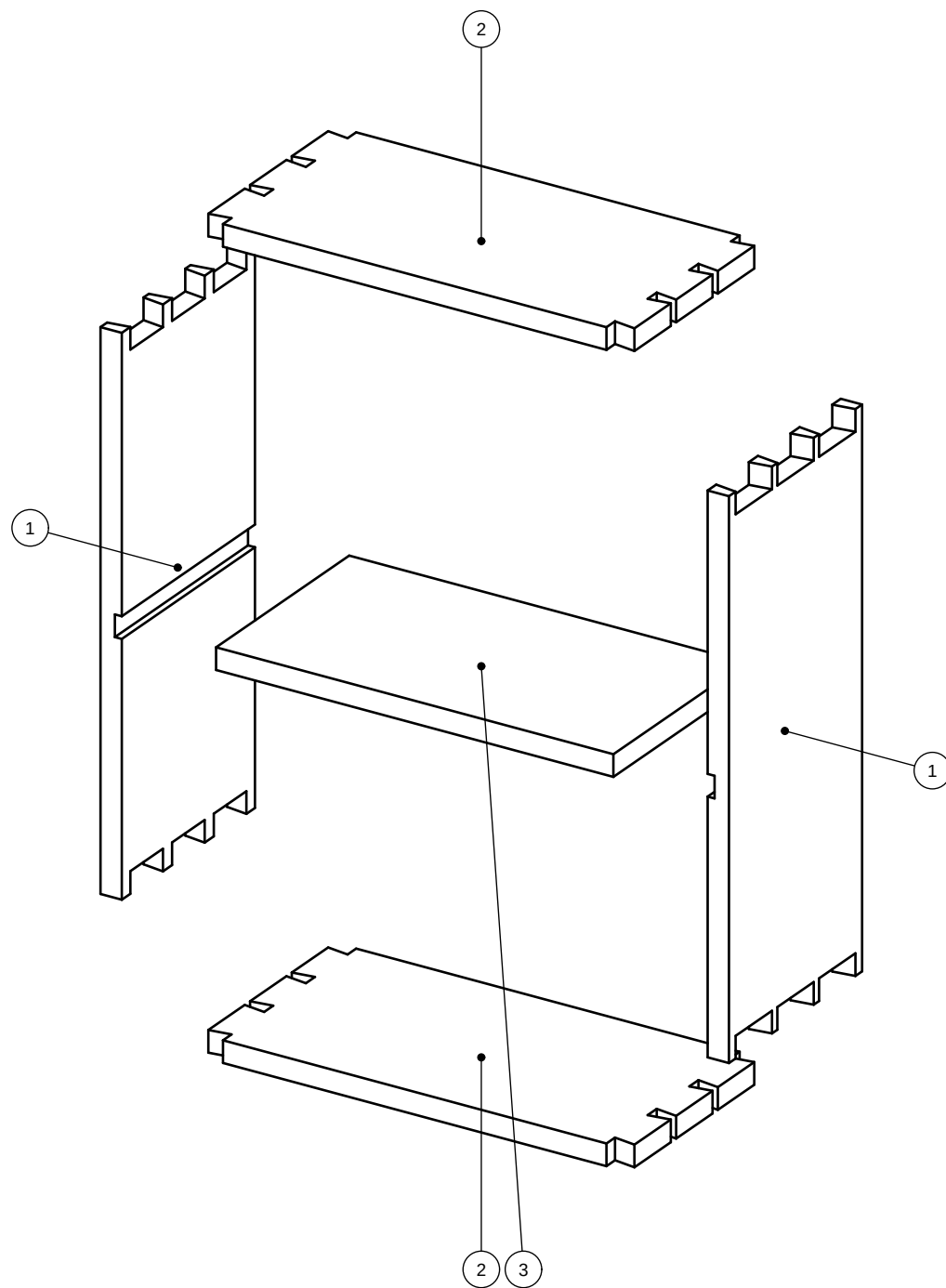


RIGHT SIDE



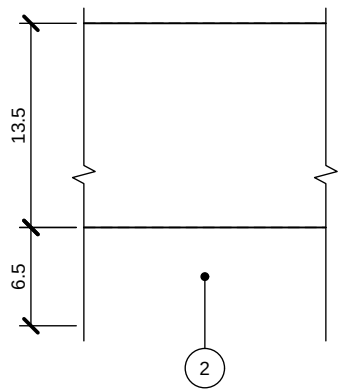
ISOMETRIC

assembly	dovetail_shelf
parts	5 (3 different)
overall	400 x 200 x 500 (W x D x H)
material	oak
scale	1:8, mm

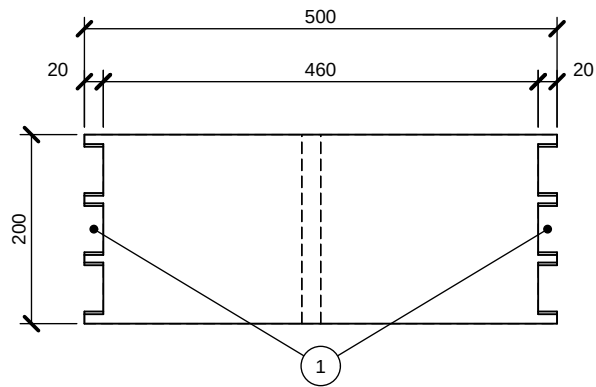


no.	part	qty	cut size (L x W x T)	material
1	left_right	2	500 x 200 x 20	oak
2	top_bottom	2	400 x 200 x 20	oak
3	shelf	1	373 x 200 x 20	oak

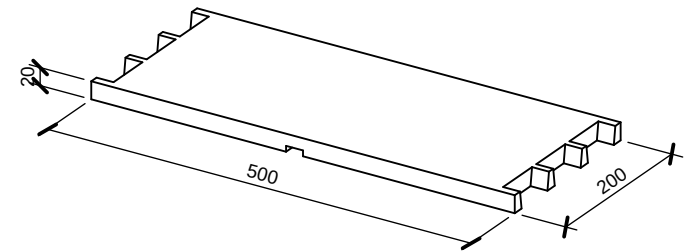
assembly	dovetail_shelf
parts	5 (3 different)
overall	400 x 200 x 500 (W x D x H)
material	oak
scale	not to scale, mm



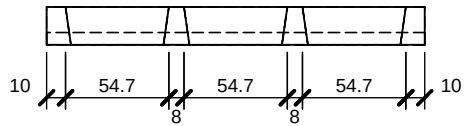
FRONT EDGE: 2 2:1



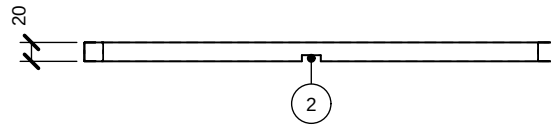
TOP FACE



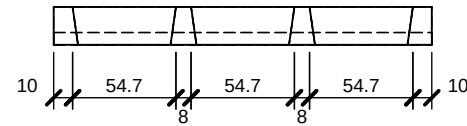
ISOMETRIC



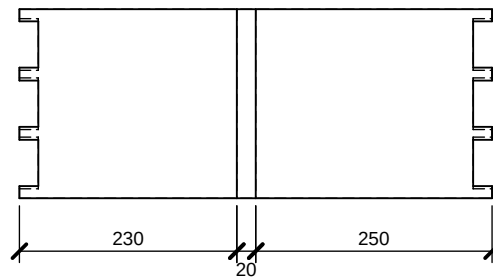
LEFT END 1:4



FRONT EDGE



RIGHT END 1:4



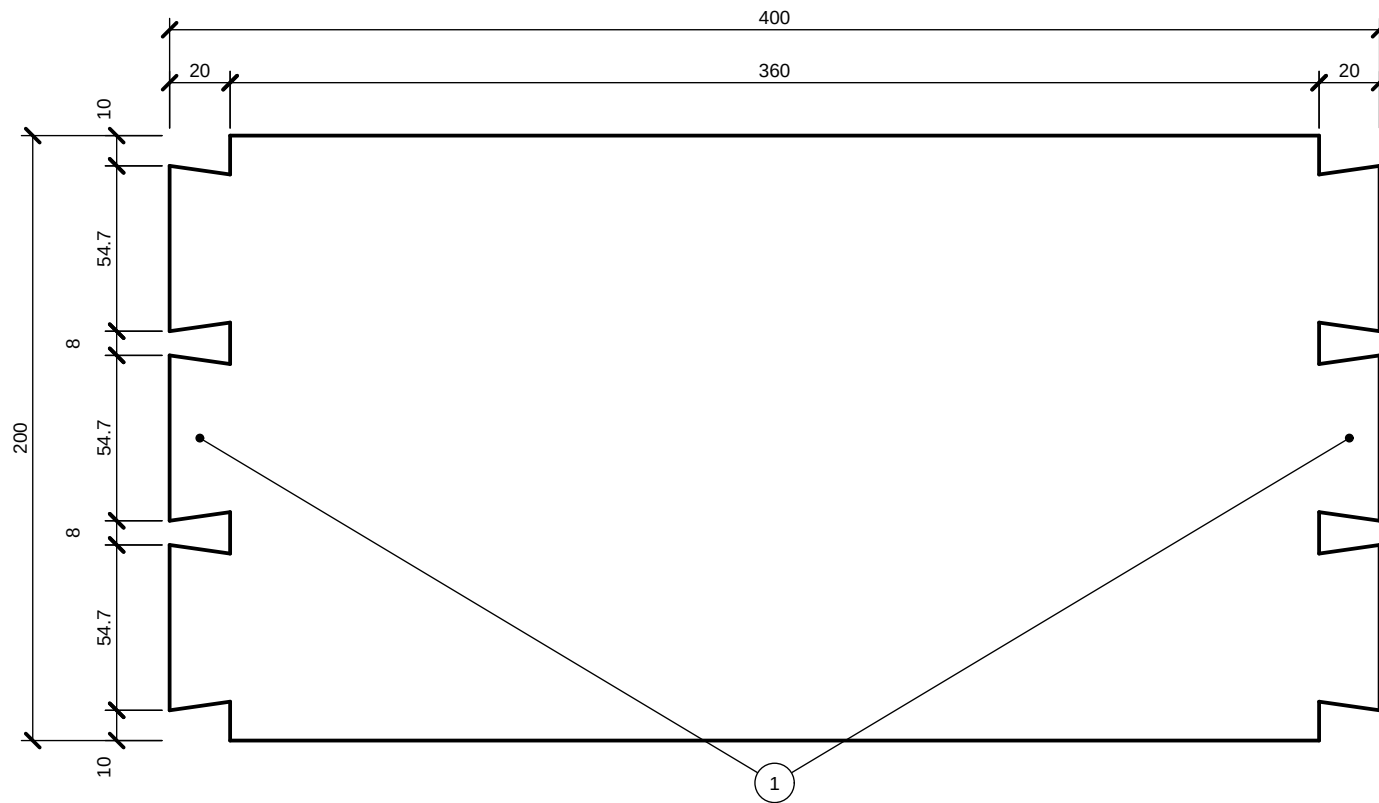
BOTTOM FACE

2 pieces: left, right

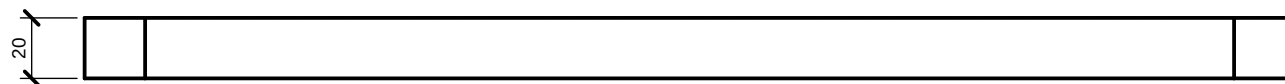
1 2 x sockets for 3 dovetails

2 dado 20 mm, 6.5 deep, in bottom_face

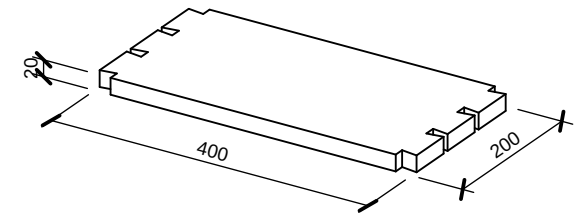
part	left_right
of	dovetail_shelf
make	2
cut size	500 x 200 x 20
material	oak
scale	1:8, mm



TOP FACE



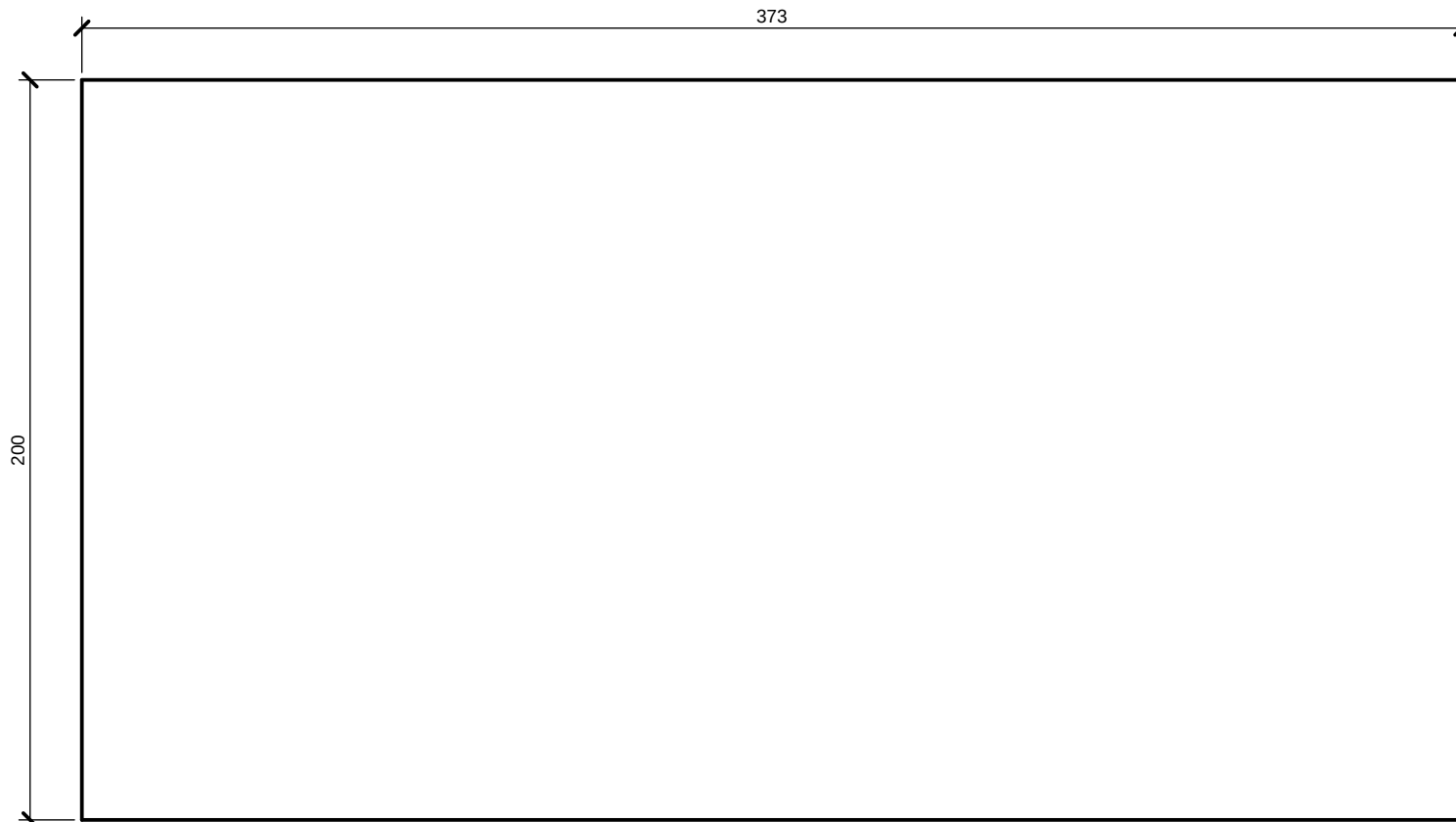
FRONT EDGE



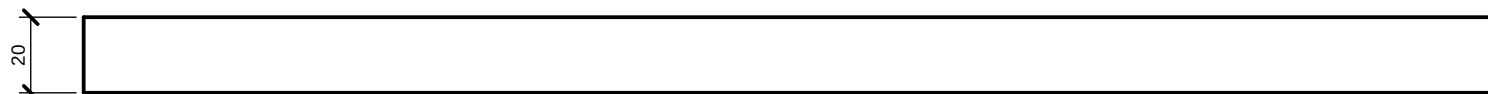
ISOMETRIC

2 pieces: top, bottom
1 2 x 3 dovetails, slope 1:7

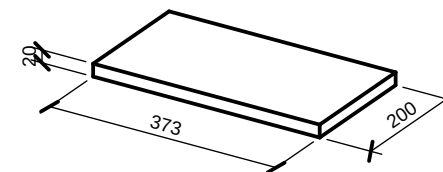
part	top_bottom
of	dovetail_shelf
make	2
cut size	400 x 200 x 20
material	oak
scale	1:2.5, mm



TOP FACE



FRONT EDGE



ISOMETRIC

part	shelf
of	dovetail_shelf
make	1
cut size	373 x 200 x 20
material	oak
scale	1:2, mm

Cut list

left_right 500 x 200 x 20 mm oak, 2 pieces: left, right
- 2 x sockets for 3 dovetails
- dado 20 mm, 6.5 deep, in bottom_face
top_bottom 400 x 200 x 20 mm oak, 2 pieces: top, bottom
- 2 x 3 dovetails, slope 1:7
shelf 373 x 200 x 20 mm oak

Appendix: source

Made with woodwright 0.9.0, build123d 0.13.0, Python 3.14.4.

To make these plans again:

```
uvx --python 3.14 woodwright==0.9.0 export projects.dovetail_shelf
```

projects.dovetail_shelf, in woodwright/examples/projects.py

```
def dovetail_shelf(
    height: float = 500,
    width: float = 400,
    depth: float = 200,
    tail_count: int = 3,
    stock: Stock = OAK,
) -> Assembly:
    """A shelf of two stories: a case through-dovetailed at its four corners
    and a middle shelf in dados; outside sizes.

    The sides have pins at both ends, the top and the bottom tails, so the
    tails show on the sides. The sides' and the top's design lengths are
    shorter by the boards they reach through.
    """
    t = stock.thickness
    asm = Assembly("dovetail_shelf", root_location=CASE_SIDE)
    left = asm.add(Board("left", stock, length=height - 2 * t, width=depth))
    right = asm.add(Board("right", stock, length=height - 2 * t, width=depth, mirror=True))
    top = asm.add(Board("top", stock, length=width - 2 * t, width=depth))
    bottom = asm.add(Board("bottom", stock, length=width - 2 * t, width=depth, mirror=True))
    shelf = asm.add(Board("shelf", stock, length=width - 2 * t, width=depth))
    for side, top_end, bottom_end, shelf_end in [
        (left, top.left_end, bottom.left_end, shelf.left_end),
        (right, top.right_end, bottom.right_end, shelf.right_end),
    ]:
        asm.join(ThroughDovetail(tails=top_end, pins=side.right_end, tail_count=tail_count))
        asm.join(ThroughDovetail(tails=bottom_end, pins=side.left_end, tail_count=tail_count))
        asm.join(
            DadoJoint(
                dado=side.bottom_face,
```

```
        end=shelf_end,  
        position=offset((height - 2 * t) / 2, side.right_end),  
    )  
    )  
    return asm
```