

Man (fog til at starte med?) skal indtaste en længde, bredde og højde, hvorefter programmet genererer en stykliste.

Hvorfor er det her vigtigt? For at vi senere kan give personen der laver tegningen nogle byggeklodser ud fra målene, som de så kan flytte rundt på og eventuelt selv tilføje byggeklodser?

Stolpelængde:

$$\text{Stolpelængde} = \text{højde cm} + 90\text{cm}$$

Antal stolper<sup>1</sup>:

$$\text{antal stolper} = \left( \frac{\text{længde cm}}{230\text{cm}} \right) \cdot 2$$

Rem-spær<sup>2</sup> længde:

$$\text{remspær længde} = \text{længde cm}$$

Antal rem-spær:

2

Spærlængde:

$$\text{spærlængde} = \text{bredde cm}$$

Antal spær til rem-spær:

$$\text{antal spær} = 1 + \frac{\text{længde cm}}{55\text{cm}}$$

Antal højrebeslag til spær:

$$\text{antal beslag} = \text{antal spær} \cdot 2$$

Antal venstrebeslag til spær:

$$\text{antal beslag} = \text{antal spær} \cdot 2$$

Hulbåndlængde:

$$\text{hulbåndlængde} = \sqrt{(\text{længde} - 55)^2 + \text{bredde}^2}$$

Antal hulbånd:

2

---

<sup>1</sup> Minimum 4 stolper? Tallet skal altid kunne deles i 2?

<sup>2</sup> Sidespær til de horisontale spær. Er samme type som spær, men længere.

Part forklaring:

String type er typen (kunne også være det unikke id?)...

Integer length er længden

int packetSize er antallet

String unitName er stk, pakke osv.

String desc er beskrivelse.

int partId er et unikt id.

(Category category er kategorien for typen)

public class PieceList:

Min hjerne vil godt have, at vi henter parts fra en database over i en liste, hvor vi så i java sætter/ændrer length og packetSize ud fra målene.

Så PieceList burde have en eller anden form for liste enten:

TreeMap<String/Integer, Part>

HashMap<String/Integer, Part>

eller

ArrayList<Part>

HashMap og TreeMap vil gøre så vi ikke behøver at lave forloops for at finde én bestemt ting ud af listen.

TreeMap vil gøre det muligt at sortere tingene i kategorier uden at vi skal slås for meget med det. Vil dog kræve et field med kategori i parts.

```
public HashMap<String/Integer, Part> getPartMap()
```

```
public void updatePartList(int length, int width, int height)
```

Alle nedenstående kunne eventuelt flyttes ind i en anden klasse:

```
private/public int numbPoles(int lengthOfCarport)
```

```
private/public int numbSideRafter(int lengthOfCarport)
```

```
private/public int numbRafters(int widthOfCarport)
```

```
private/public int numbRightMounts(int numberOfRafters)
```

```
private/public int numbLeftMounts(int numberOfRafters)
```

```
private/public int numbTrapezRoof(int lengthOfCarport, int widthOfCarport)
```

```
private/public int lengthOfPole(int heightOfCarport)
```

```
private/public int lengthOfRafter(int widthOfCarport)
```

```
private/public int lengthOfMetalTape(int lengthOfCarport)
```

```
private/public int numberOfMetalTape(int length of metalTape)
```

osv.

```
public class PartMapper
```

```
public HashMap<String/Integer, Part> getStandardParts()
```

```
public Part getPart(String/Integer uniquePartId)
```

```
public void addPartList(HashMap<String/Integer, Part> partMap, User user)
```