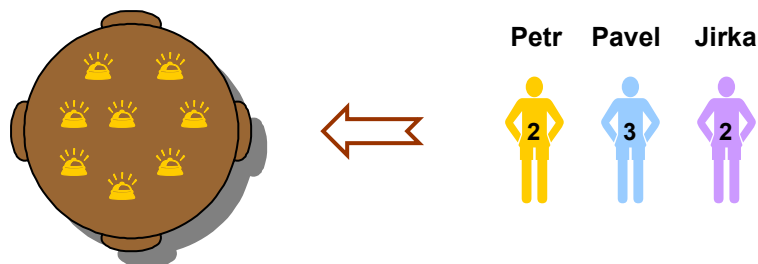


3. Jedno z předchozích zadání (1. nebo 2.) implementujte i **alternativním způsobem** (tj. **frontu seznamem** nebo **zásobník dynamickým polem**) a ověřte, že klientskou třídu nemusíte změnit.
4. Na stole je talíř a na něm je p chutovek. Okolo sedí n strážníků, kteří ke svému nasycení potřebují sníst k_i ($k_i \geq 0$) chutovek ($i \in \langle 1, n \rangle$). V zájmu spravedlnosti se strážníci v jídle střídají jeden po druhém a každý sní vždy jen jednu chutovku. Najedí se všichni? Kdo zůstane hladový?

Tuto úlohu implementujte pomocí fronty (pro libovolný počet strážníků). Strážníci jsou reprezentováni položkami třídy `Stravnik` s atributy jméno strážníka a velikost jeho hladu, který je reprezentován počtem chutovek potřebných k nasycení. Vyberte strážníka z fronty, uberte jednotku z velikosti hladu strážníkovi voláním metody `snezChutovku()` a jednotku z počtu chutovek na talíři. Není-li strážník nasycen, vložte jej zpět do fronty.

Přečtěte nejprve počet strážníků a množství jídla reprezentované počtem chutovek a pak jména strážníků a velikosti jejich hladu. Pak **postupně vypisujte jména sytých strážníků**. V případě, že jsou všichni nasyceni, vypište **množství zbylého jídla** (počet chutovek). Pokud se všichni nenajedli, vypište **jména nenasyčených strážníků a velikosti jejich "zbylého" hladu**.

Důležité: Se strážníky ve frontě pracujte výhradně pomocí operací rozhraní ADT fronta, tedy operací `vloz()`, `vyber()` a `jePrazdna()`.



Vzor vstupu 1:

3
8
Petr 2
Pavel 3
Jirka 2

Vzor výstupu 1:

Syti:
Petr
Jirka
Pavel
Zbylo 1 chutovek

Vzor vstupu 2:

3
8
Petr 3
Pavel 4
Jirka 2

Vzor výstupu 2:

Syti:
Jirka
Petr
Hladovi:
Pavel ma hlad na 1 chutovek