Person

• has a name, a reference to a fiancée, a reference to a Queue of names ordered by preference, and a reference to a Map of names to rankings

Note that "rankings" contains similar information to "preferences" but in a different format. Use either one depending on what you need.
Your constructor is passed two Sets of Person objects:

**men**

- **George**
  - fiancee = null
  - preferences = ...
  - rankings = ...

- **Kramer**
  - fiancee = null
  - preferences = ...
  - rankings = ...

- **Jerry**
  - fiancee = null
  - preferences = ...
  - rankings = ...

- **Newman**
  - fiancee = null
  - preferences = ...
  - rankings = ...

**women**

- **Samantha**
  - fiancee = null
  - preferences = ...
  - rankings = ...

- **Miranda**
  - fiancee = null
  - preferences = ...
  - rankings = ...

- **Carrie**
  - fiancee = null
  - preferences = ...
  - rankings = ...

- **Charlotte**
  - fiancee = null
  - preferences = ...
  - rankings = ...
• Your MatchMaker should create a map from names to Persons

Creating a map makes it easy for you to instantly look up a person by name instead of having to search through the sets.
How do I know whether Miranda prefers Jerry or George?

1. Get Miranda from your Map
2. Get her rankings of George and Jerry from her Map
3. The lower ranking is preferred
Some pseudocode

• Suppose you have a `Person` variable named `p1` referring to the `Person` object representing George. How do you find out George's most preferred woman?
  – Look at the front of `p1`'s `getPreferences` queue.

• Suppose George's most preferred woman is currently Miranda, and that you have a `Person` variable named `p2` referring to the `Person` object representing Miranda. How do we find out how well Miranda likes George?
  – Look in `p2`'s `getRankings` map.
  – If you give this map a string (a name) as a key, it will give you an integer ranking as a value.
  – That integer is Miranda's ranking for George (1=best, 4=worst).

• How do I get this `Person` variable `p2` to find Miranda?
  – Your map can help you find a `Person` object by name.