Harry Potter

After the tragic death of young Harry Potter's parents, he spends his daily life with his hateful uncle. However, his life changes in one fell swoop when a giant comes to him to tell him he is a wizard and must continue his studies at Hogwarts School of Witchcraft and Wizardry. The school has four houses competing with each other. Students are assigned to a house by the Sorting Hat based on the young student's qualities: the brave and reckless wizards go to Gryffindor, the eminently clever and wise to Ravenclaw, the persistent and loyal to Hufflepuff, and the cunning and ambitious to Slytherin.

- Create youthful wizards who are randomly assigned their qualities (Courage, Intelligence, Perseverance, Cunning). Give them a name as well (this can be randomly generated from a predefined first name / last name set).
- Create each house where the wizards are stored in a binary search tree. The key of the search tree should always be the trait that that house values best (e.g., courage in the case of Gryffindor).
- Write a program that helps the Sorting Hat assigning wizards and witches to each house so that each house has as many wizard apprentices as possible that have the highest value of the property corresponding to that house. Also, make sure that you get nearly the same amount of wizards in each house.
- After the sorting procedure, the user should be able to query which wizards have some property *P* with a value at least *V*, by listing which house each wizard belongs to.

Example:

```
Input:

  Courage 5

Output:

  Gryffindor: John Doe - Harry Potter - Ron Weasley

  Ravenclaw: Allsaint Alice

  Slytherin: Draco Malfoy – Bloody Bob
```

Everything that is not specified here can be implemented using any approach respecting common logic and the object-oriented principles. Loosely couple your classes using interfaces, events and exceptions.