

## 4. Forskellige spillere

Formålet med denne opgave er at lave forskellige typer af spillere, som kan deltage i en turnering. En deltager i turneringen kan være en enkelt spiller, en juniorspiller eller et hold med flere spillere. Alt efter typen af deltager, bliver point fordelt forskelligt.

1. Lav et interface `Player`
2. Giv interfacet metoderne `getPoint()`, `getName()` og `addPoints()`. Du bestemmer selv parametre og returntype. Læs resten af opgaven for at få en idé om hvilke datatyper, det vil give mening at bruge.
3. Lav en klasse `StandardPlayer`. Lad klassen implementere `Player`.
  - a. Giv klassen relevante attributter og konstruktør.
  - b. Implementer metoderne `getName()` og `getPoints()`.
  - c. Implementer metoden `addPoints()`. En standardspiller får tilføjet point uden særlige regler. Det vil sige, at hvis spilleren modtager 10 point, bliver der lagt 10 point til vedkommendes samlede antal point.
  - d. Override metoden `toString()`, så den returnerer "Name: <navn>, points: <points>" hvor det, der står i <> er navnet og antal point.
4. Lav en klasse `JuniorPlayer`. Lad klassen implementere `Player`.
  - a. Giv klassen relevante attributter og konstruktør.
  - b. Implementer metoderne `getName()` og `getPoints()`.
  - c. Implementer metoden `addPoints()`. En juniorspiller får halvanden gange point i forhold til standardspilleren. Det vil sige, at hvis juniorspilleren modtager 10 point, bliver der lagt 15 point til vedkommendes samlede point.
  - d. Override metoden `toString()`, så den returnerer "Name: <navn>, points: <points>" hvor det, der står i <> er navnet og antal point.
5. Lav en klasse `Team`. Lad klassen implementere `Player`.
  - a. Et team består af flere spillere, som kan være standard- eller juniorspillere. Lav en attribut `players`, som er en samling af spillere.
  - b. Giv klassen eventuelle andre relevante attributter, samt en konstruktør, som sætter værdien af `players`.
  - c. Implementer metoden `getName()`.
  - d. Implementer metoden `addPoints()`. Et team fordeler sine point blandt spillerne. Det vil sige, at hvis der er fire spillere på et hold og holdet får 10 point, vil hver spiller modtage 2,5 point.
  - e. Implementer metoden `getPoints()` så spillernes point tælles sammen og returneres.
  - f. Override metoden `toString()`, så den returnerer "Name: <navn>, points: <points>" hvor det, der står i <> er navnet på teamet og spillernes samlede antal point.
6. Lav en klasse `Main` med en `main`-metode.
  - a. Lav nogle objekter af dine klasser og test at din kode virker som forventet.

Hvis du har tid:

Lav en klasse `Tournament`, som har en samling af flere spillere. Lav metoden `getTopTen()` som returnerer de ti spillere med flest point (evt. sorteret efter antal point). Lav metoden `getAverage()` som returnerer det gennemsnitlige antal point.