

R4.04 : Qualité de développement

Observateur

Station météo

Une station météo enregistre des mesures de température et d'humidité. Elle est modélisée par la classe suivante :

```
export class StationMétéo {
  private _temperature: number;
  private _humidité: number;

  constructor(temperature = 0, humidité = 15) {
    this._temperature = temperature;
    this._humidité = humidité;
  }

  public get temperature(): number {
    return this._temperature;
  }

  public set temperature(temperature: number) {
    this._temperature = temperature;
  }

  public get humidité(): number {
    return this._humidité;
  }

  public set humidité(humidité: number) {
    if (humidité < 0 || humidité > 100) {
      throw new Error("L'humidité est exprimée en pourcentage !");
    }
    this._humidité = humidité;
  }

  public toString(): string {
    return this._temperature + "°C\n" + this._humidité + "%";
  }
}
```

Mise en place d'un affichage

On désire mettre en place deux types d'affichage à chaque fois que la station météo est modifiée :

- un affichage console simple qui affiche la température et l'humidité sur deux lignes

- un affichage console plus élaboré qui affiche une courbe contenant à la fois les températures et l'humidité

Affichage texte

Cet afficheur se contente d'afficher la dernière température et le dernier taux d'humidité reçus. Voici une ébauche de cette classe :

```
export class AfficheurTexte {
  affiche(temperature: number, humidité: number): void {
    console.log(temperature + "°C\n" + humidité + "%");
  }
}
```

Affichage graphique

Cette classe conserve les données reçues à chaque notification, pour afficher la courbe à chaque modification. Un squelette de cette classe est donné ci-dessous :

```
import * as asciichart from "asciichart";

export class AfficheurGraphique {
  private tabTemperature: Array<number>;
  private tabHumidité: Array<number>;

  constructor() {
    this.tabTemperature = [];
    this.tabHumidité = [];
  }

  affiche(): void {
    console.log(asciichart.plot([this.tabTemperature, this.tabHumidité]));
  }
}
```

Indication : vous aurez besoin d'installer les modules `asciichart` et `@types/asciichart` pour pouvoir utiliser cette classe.

```
npm install asciichart @types/asciichart
```

Objectif

On dispose de l'ébauche d'application suivante :

```

import { StationMétéo } from "./station-meteo";
import { AfficheurTexte } from "./afficheur-texte";
import { AfficheurGraphique } from "./afficheur-graphique";

function main(): void {
    const aff = new AfficheurTexte();
    const aff2 = new AfficheurGraphique();
    const stationMeteo = new StationMétéo(10);

    console.log("Appuyez sur entrée pour le prochain affichage");
    rs.question();

    stationMeteo.humidité++;
    for (let p = 0; p < 10; p++) {
        console.log("Appuyez sur entrée pour le prochain affichage");
        rs.question();

        stationMeteo.humidité--;
        stationMeteo.temperature += 2;
    }

    for (let p = 0; p < 10; p++) {
        console.log("Appuyez sur entrée pour le prochain affichage");
        rs.question();

        stationMeteo.humidité++;
        stationMeteo.temperature -= 3;
    }
}

main();

```

On désire mettre en place un patron de conception de type observateur pour que les deux afficheurs soient prévenus à chaque modification de la station météo et affiche l'information. L'exécution de l'application doit donner le résultat suivant (ou équivalent) :

Appuyez sur entrée pour le prochain affichage

10°C

16%

```

16.00 +
15.00 -
14.00 -
13.00 -
12.00 -
11.00 -
10.00 +

```

Appuyez sur entrée pour le prochain affichage

10°C

15%

16.00 |
15.00 |
14.00 |
13.00 |
12.00 |
11.00 |
10.00 |

12°C

15%

16.00 |
15.00 |
14.00 |
13.00 |
12.00 |
11.00 |
10.00 |

Appuyez sur entrée pour le prochain affichage

12°C

14%

16.00 |
15.00 |
14.00 |
13.00 |
12.00 |
11.00 |
10.00 |

14°C

14%

16.00 |
15.00 |
14.00 |
13.00 |
12.00 |
11.00 |
10.00 |

Appuyez sur entrée pour le prochain affichage

...

Travail à faire

1. Mettre en place le patron de conception en vous basant sur les classes et programmes fournis ci-dessus. (les fichiers sont sur ARCHE)
2. Donner le diagramme de classes UML de votre solution