# TimeTrack

Eine Firma benötigt ein Zeiterfassungssystem für ihre Mitarbeiter:innen. Es sollen die Arbeitszeit, Krankenstände und Urlaube eingetragen werden. Die Klassen TimeTrack und TimeSheet sind die Kernkomponenten des Systems.

Hier die Schnittstellen mit Beschreibung der Klasse TimeSheet.

```
package oop.classes;

public class TimeSheet {

    /* private fields here */

    /*
            Creates a new instance of TimeSheet for the
            specified employee id.
     */
    public TimeSheet(int employeeId) {
        ...
    }

    /*
            Creates a new instance of TimeSheet that is
            a deep copy of the input argument.
            (copy constructor)
     */
    public TimeSheet(TimeSheet copyFrom) {
        ...
    }

    /*
            Simple setter
     */
    public void setWorkingHours(int workingHours) {
        ...
    }

    /*
            Simple setter
     */
    public void setSickLeave(int sickLeave) {
        ...
    }

    /*
            Simple setter
     */
    public void setVacation(int vacation) {
        ...
    }

    /*
            Simple getter
    */
    public int getEmployeeId() {
```

```
        ...
    }

    /*
            Simple getter
     */
    public int getSickLeave() {
        ...
    }

    /*
            Simple getter
     */
    public int getWorkingHours() {
        ...
    }

    /*
            Simple getter
     */
    public int getVacation() {
        ...
    }

    @Override
    public String toString() {
        ...
    }
}
```

Hier die Schnittstellen der Klasse TimeTrack mit Beschreibungen.

```
package oop.classes;

public class TimeTrack {

    /* private fields */

    /*
            Creates a new instance of TimeTrack by passing in the
            id of the affected employee.
     */
    public TimeTrack(int employeeId) {
        ...
    }

    /*
            Creates a new instance of TimeTrack that is initialized
            with the passed in TimeSheet instance.
     */
    public TimeTrack(TimeSheet timeSheet) {
        ...
    }

    /*
       Gets a deep copy of the current timesheet.
```

```
        The timesheet contains the currently added
        time tracking data (e.g.: workinghours, sickleave)
    */
    public TimeSheet getCurrentTimeSheet()  {
        ...
    }

    /*
            Adds the specified amount of working hours to the time tracking.
            After each 6 hours working time a break of 1 hour has to be made.
            The required break time is subtracted from the passed working hours
            automatically (e.g.: 14 working hours -> 2 hours for break are subtracted
            -> 12 working hours are effectively added).
    */
    public void addWorkingHours(int hours) {
        ...
    }

    /*
            Registers the specified amount of days as sick leave.
     */
    public void addSickLeave(int days) {
        ...
    }

    /*
            Add the specified amount of vacation days.
     */
    public void addVacation(int days) {
        ...
    }

    /*
            Starts a new time tracking period and resets
            all already stored tracking data.
     */
    public void startNewPeriod() {
        ...
    }
}
```

Implementieren sie die beiden Klassen, sodass die in den Kommentaren beschriebene Schnittstellenlogik erfüllt ist. Beispielsweise sollte folgende Verwednung möglich sein:

```
package oop.classes;

public class Main {
    public static void main(String[] args) {
        int employeeId = 2234;
        var timeTrack = new TimeTrack(employeeId);
        System.out.println(String.format("Create timesheet of Employee with id %d", employeeId));
        System.out.println(String.format("Add %d working hours.", 10));
        timeTrack.addWorkingHours(10);
        System.out.println(String.format("Add another %d working hours.", 12));
        timeTrack.addWorkingHours(12);
        System.out.println(String.format("Add %d days sick leave.", 1));
```

```
        timeTrack.addSickLeave(1);
        System.out.println(String.format("Add another %d days sick leave", 1));
        timeTrack.addSickLeave(1);
        System.out.println(String.format("Add %d days sick leave.", 2));
        timeTrack.addVacation(2);
        System.out.println(String.format("Add %d days sick leave.", 1));
        timeTrack.addVacation(1);
        TimeSheet timeSheet = timeTrack.getCurrentTimeSheet();
        System.out.println(timeSheet.toString());
    }
}
```

Folgende Ausgabe erscheint in der Konsole beim Durchlauf von Main:

```
Create timesheet of Employee with id 2234
Add 10 working hours.
Add another 12 working hours.
Add 1 days sick leave.
Add another 1 days sick leave
Add 2 days sick leave.
Add 1 days sick leave.
Time sheet of employeeId -> 2234
   workingHours -> 19
   sickLeave -> 2
   vacation -> 3
```