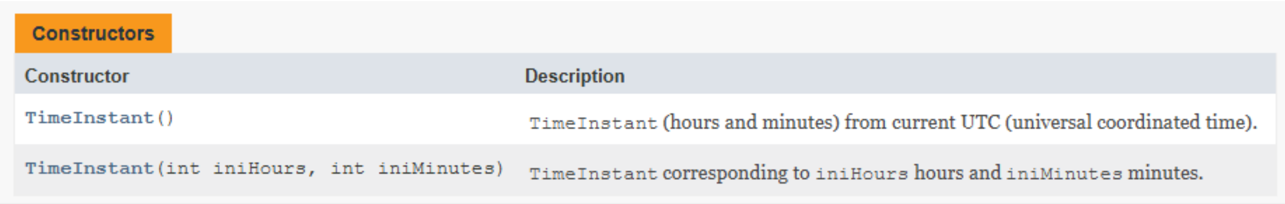
First mid term exam of IIP – ETSINF – October 31^{st}, 2019 – Duration 90 minutes

Note: The maximum mark of this exam is 10 points, but its weight in the final grade of IIP is **3.75 points**

NAME:	GROUP:

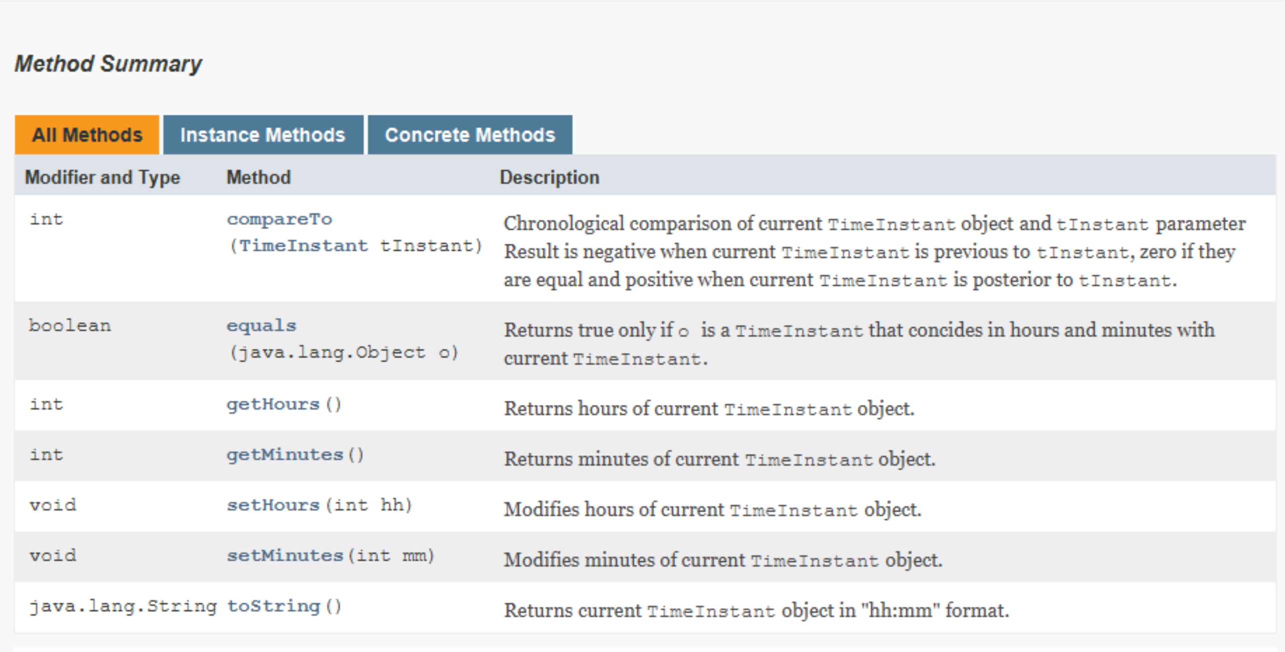
1. $\boxed{\text{6 points}}$ A data-type class must be designed in order to represent news to be published on digital media. The name of the class must be `PieceOfNews`. Every single news will represented as an instance of this class and will have the following attributes: (a) time the news was generated, (b) link to the file containing the information to be published, (c) number of media that have echoed the news in the same day, and (d) news type: text, audio or video.

It is available the class `TimeInstant` to represent the moment in the day each news was generated. Next you have the relevant parts of the documentation of the class `TimeInstant` needed for the exam:



Constructors

Constructor	Description
`TimeInstant()`	`TimeInstant` (hours and minutes) from current UTC (universal coordinated time).
`TimeInstant(int iniHours, int iniMinutes)`	`TimeInstant` corresponding to `iniHours` hours and `iniMinutes` minutes.



Method Summary

Modifier and Type	Method	Description
`int`	`compareTo (TimeInstant tInstant)`	Chronological comparison of current `TimeInstant` object and `tInstant` parameter Result is negative when current `TimeInstant` is previous to `tInstant`, zero if they are equal and positive when current `TimeInstant` is posterior to `tInstant`.
`boolean`	`equals (java.lang.Object o)`	Returns true only if `o` is a `TimeInstant` that concides in hours and minutes with current `TimeInstant`.
`int`	`getHours()`	Returns hours of current `TimeInstant` object.
`int`	`getMinutes()`	Returns minutes of current `TimeInstant` object.
`void`	`setHours(int hh)`	Modifies hours of current `TimeInstant` object.
`void`	`setMinutes(int mm)`	Modifies minutes of current `TimeInstant` object.
`java.lang.String`	`toString()`	Returns current `TimeInstant` object in "hh:mm" format.

What you have to do: implement the class `PieceOfNews` with the following attributes and methods (you can assume this class will be in the same package the class `TimeInstant` is):

a) (0.5 points) Three constants that will public and static attributes of type `int`. The purpose is to use a numeric code for each one of the three types of news, this will allow to distinguish the data format of the file containing the news. The names of the three constants must be `AUDIO`, `VIDEO` and `TEXT` with values 0, 1 and 2 respectively.

This constants must be used wherever in the code of the methods of the class `PieceOfNews` when the type of the news needs to be specified. Never use the values of the constants directly.

b) (0.5 points) Four private attributes (i.e. instance variables), to represent the elements of a news described above. Use the identifiers proposed in this list: `instant` (`TimeInstant`); `link` (`String`); `echoedBy` (`int`); `type` (`int`).

c) (0.75 points) Constructor to create objects of the class `PieceOfNews` to represent a news generated at time `i`, stored in the file referenced by the link `l`, echoed by a total of `n` media, and with news type `t`. In order to simplify the implementation, you can assume that the values of the parameters will always be correct.

d) (1.25 points) The method `equals` for overriding the method with the same name inherited from the class `Object`. This method must check if the news represented by the current object (`this`) is the same than other provided as a parameter. First, the method must check if the provided object is an instance of the class `PieceOfNews`, then check if both were generated at the same time, have been echoed by an equal number of media, and have the same news type. The link should not be taken into account.

e) (1.75 points) The method `compareTo` to compare two objects of the class `PieceOfNews`, as in the equals method, one news is the current object (`this`), and the other is provided as parameter, e.g. `other`. The comparison must be done according to the popularity, in such a way that if `this` is less popular than `other` returns a negative integer, if `this` is more popular than `other` returns a positive integer, otherwise returns 0 for indicating that both news are equally popular.

The popularity criteria are the following ones:

- Any news is less popular than another if was generated before.

- If both news were generated at the same time, then one news is less popular than another if was echoed by less media than the other.
- In the case both news where generated a the same time and echoed by the same number of media, then the popularity is conditioned by the degree of elaboration, that is, the news type. So, news in text written by a journalist are the most elaborate, then videos, and audio files are considered the less elaborate.

f) (1.25 points) The `toString` method that must override the method with the same name inherited from the class `Object`. This method returns a description of the news in the following format: the time it was generated, the link, the number of media that echoed it, and the news type with one of the following three words between parenthesis: `text`, `video`, `audio`. Here you have an example:

```
10:30 https://media.com/2019/10/31/climate-change2 150 (text)
```

Solution:

```
public class PieceOfNews {
    public static final int AUDIO = 0, VIDEO = 1, TEXT = 2;

    private TimeInstant instant;
    private String link;
    private int echoedBy;
    private int type;

    public PieceOfNews(TimeInstant i, String l, int n, int t) {
        instant = i;
        link = l;
        echoedBy = n;
        type = t;
    }

    public boolean equals(Object o) {
        return o instanceof PieceOfNews
            && this.instant.equals(((PieceOfNews) o).instant)
            && this.echoedBy == ((PieceOfNews) o).echoedBy
            && this.type == ((PieceOfNews) o).type;
    }

    public int compareTo(PieceOfNews other) {
        int res = this.instant.compareTo(other.instant);
        if (res == 0) {
            res = this.echoedBy - other.echoedBy;
            if (res == 0) {
                res = this.type - other.type;
            }
        }
        return res;
    }

    public String toString() {
        String res = "";
        res += instant + " " + link + " " + echoedBy + " (";
        switch (type) {
            case TEXT:
                res += "text)"; break;
            case VIDEO:
                res += "video)"; break;
            default:
                res += "audio)";
        }
        return res;
    }
}
```

2. 2 points **What you have to do**: given the next program-type class `TestPieceOfNews`, complete the `main` method to perform the actions described next. For simplifying, you can assume that this class will be in the same package the classes `PieceOfNews` and `TimeInstant` are. In this class you have to use the constants defined in the classes of the same package wherever be required.

```
public class TestPieceOfNews {
    /** Returns a random integer in the rang [start, end], 0 <= start < end. */
    private static int random(int start, int end) {
        return (int) (Math.random() * (end - start + 1) + start);
    }

    public static void main(String[] args) {
        ...
    }
}
```

a) (0.25 points) Create an object of the class `TimeInstant` named `ti` to represent 10:30 AM.

b) (0.25 points) Create an object of the class `PieceOfNews` named `n1` to represent a news generated at time `ti`, of type audio, echoed by a total of `200` media, and whose link is `"https://media.com/2019/10/31/climate-change1"`.

c) (0.25 points) Assign to a local variable named `echo2` a random number in the range $[2, 500]$.

d) (0.25 points) Create an object of the class `PieceOfNews` named `n2` generated at time `ti`, of type text, that has been echoed by a total of `echo2` media, and whose link is `"https://media.com/2019/10/31/climate-change2"`.

e) (0.25 points) Assign to a local variable named `resC` the result of executing the method `compareTo` with respect to the object referenced by `n1` in order to compare the both news referenced by the local variables `n1` y `n2`.

f) (0.75 points) Depending on the value of `resC`, show on the screen the most popular news by using the format generated by the method `toString`. In the case both news were equally popular, then show both news on the screen separated by the equal sign.

Solution:

```
public class TestPieceOfNews {

    /** Returns a random int value in the range [ini, fin], 0 <= ini < fin. */
    private static int random(int ini, int fin) {
        return (int) (Math.random() * (fin - ini + 1) + ini);
    }

    public static void main(String[] args) {
        TimeInstant ti = new TimeInstant(10, 30);
        String link = "https://media.com/2019/10/31/climate-change1";
        PieceOfNews n1 = new PieceOfNews(ti, link, 200, PieceOfNews.AUDIO);

        int echo2 = random(2, 500);
        String link2 = "https://media.com/2019/10/31/climate-change2";
        PieceOfNews n2 = new PieceOfNews(ti, link2, echo2, PieceOfNews.TEXT);

        int resC = n1.compareTo(n2);
        if (resC == 0) {
            System.out.println(n1 + " = " + n2);
        }
        else if (resC < 0) {
            System.out.println(n2);
        }
        else { System.out.println(n1); }
    }
}
```

3. 2 points Given the following class where the class `TimeInstant` is used, what is shown on screen when executed?

```
public class Exercise3 {

    public static void main(String[] args) {
        TimeInstant aux = new TimeInstant(5, 6); int j = 1;
        System.out.println("At main: " + j + " " + aux.getHours() + " " + aux.getMinutes());
        m2(aux, j);
        System.out.println("At main: " + j + " " + aux.getHours() + " " + aux.getMinutes());
    }

    private static void m2(TimeInstant aux, int j) {
        System.out.println("At m2:   " + j + " " + aux.getHours());
        int newHour = aux.getHours() + j;
        j++; aux.setHours(newHour);
        m1(aux, j);
        System.out.println("At m2:   " + j + " " + aux.getHours());
    }

    private static void m1(TimeInstant aux, int j) {
        System.out.println("At m1:   " + j + " " + aux.getMinutes());
        int newMinutes = aux.getMinutes() + j;
        j++; aux.setMinutes(newMinutes);
        System.out.println("At m1:   " + j + " " + aux.getMinutes());
    }
}
```

Solution:

```
At main: 1 5 6
At m2:   1 5
At m1:   2 6
At m1:   3 8
At m2:   2 6
At main: 1 6 8
```