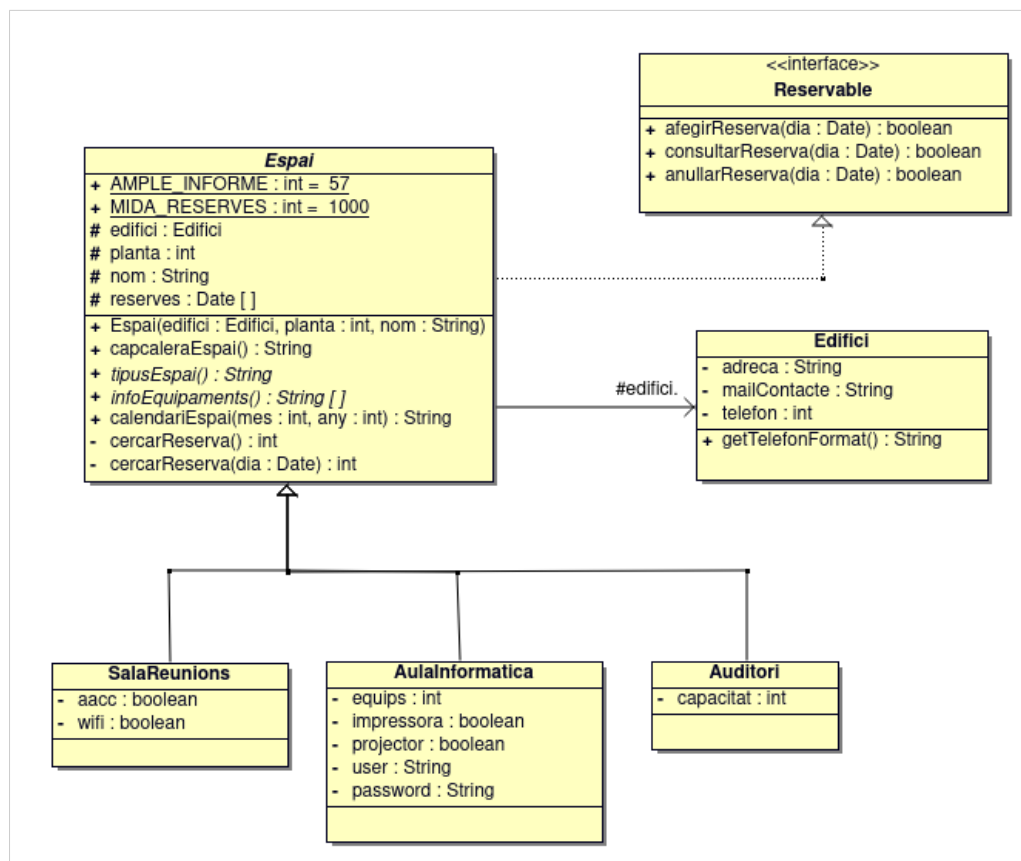




## DAM2/DAW2 – Pe1 2023-24

### Diagrama de classes UML



### Enunciat

Implementeu el diagrama de classes de l'esquema amb els atributs i mètodes indicats. Adjunt a l'enunciat trobareu un programa amb dades de prova «**MainUF4Pe1**» que quan acabeu hauria de generar una sortida com la que es mostra a l'[annex A](#).

### Consideracions

- Respecteu **els noms i les definicions** de les classes, interfícies, atributs, mètodes i paràmetres de l'esquema.
- Cal que preueu atenció a:
  - Les **relacions** entre les classes i/o interfícies de l'esquema: associacions, herències, implementacions
  - Els diferents **modificadors** dels atributs i mètodes («private», «public», «protected», «static»)
  - **Formats** que aportin altra informació rellevant: cursiva, subratllat.



- **Valors** de les constants
- Els **«getters/setters»** no es mostren a l'esquema per simplificar però mentre no s'indiqui el contrari cal crear aquests mètodes per a tots els atributs que no siguin accessibles públicament.
- Per les classes on no s'indica cap constructor cal afegir-ne un amb tots els atributs.
- Podeu afegir **noves constants** públiques o privades a les classes sempre que us calgui. També podeu afegir altres **mètodes privats** (no públics) que us puguin ser d'utilitat.
- Finalment cal que garantiu en la mesura del possible la **coherència de les dades** afegint les validacions que cregueu oportunes. Tant alhora de crear les instàncies com al modificar-les.

## Requeriments

### Interfície «Reservable»

- `afegirReserva(dia): boolean` Afegir una reserva per aquest dia, retorna si és possible fer la reserva.
- `consultarReserva(dia): boolean` Consultar si hi ha una reserva per aquest dia.
- `anullarReserva(dia): boolean` Anul·lar una reserva per aquest dia, retorna si és possible fer l'anul·lació.

### Classe «Edifici»

- «adreça»: adreça de l'edifici, no pot ser nul (substituir per un text buit).
- «mailContacte»: adreça de correu de contacte de l'edifici, no pot ser nul·la (substituir per un text buit).
- «telefon»: telèfon de contacte de l'edifici, ha de començar per 6 o per 9<sup>1</sup> i tenir exactament 9 dígits (substituir per 999999999).

- `getTelefonFormat(): String` Retorna un text amb la representació del telèfon, format: XXX XX XX XX

### Classe abstracte «Espai» implementa «Reservable»

- `«MIDA_RESERVES»` (constant): mida del vector de reserves.
- `«AMPLE_INFORME»` (constant): Amplada en caràcters del calendari de reserves dels espais.
- «edifici»: edifici on es troba l'espai, no pot ser nul.
- «planta»: planta del edifici on s'ubica l'espai.
- «nom»: nom de l'edifici, no pot ser nul (substituir per un text buit).
- «reserves»: vector amb els dies que l'espai està reservat, la mida d'aquest vector està indicada a la constant `«MIDA_RESERVES»`.

- `Espai(edifici, planta, nom)` Constructor inicialitza les dades que arriben des dels paràmetres i el vector de reserves que inicialment està buit.

<sup>1</sup> Per fer la validació podeu consultar els mètodes de la classe «StringUtils».



`capcaleraEspai(): String`

Retorna el text de capçalera del calendari de l'espai:

- L'adreça de l'edifici i dades de contacte, tipus d'espai, nom i planta on s'ubica.
- La informació dels equipaments de l'espai.

**Aquest mètode s'indica a l'[Annex B](#).**

`tipusEspai(): String`

Mètode abstracte. Les implementacions han de retornar un text indicant el nom de l'espai.

`infoEquipaments(): String[]`

Mètode abstracte. Les implementacions han de retornar un text descriptiu de cada recurs o equipament de l'espai.

`calendariEspai(mes, any):String`

Retorna el calendari en format de graella setmanal pel mes i any indicats. Pels dies disponibles (sense reserva) es mostra el número del dia del mes, pels dies reservats es mostra «XX», i pels dies a l'inici o al final que no pertanyen al mes es mostra «-».

**Aquest mètode s'indica a l'[Annex B](#).**

`afegirReserva(dia): boolean`

Permet afegir la reserva per aquest dia sempre que sigui estrictament posterior a la data actual. Retorna si es pot fer la reserva o no.

`consultarReserva(dia): boolean`

Retorna si hi ha una reserva per aquest dia o no.

`anullarReserva(dia): boolean`

Permet anul·lar una reserva per aquest dia sempre que des de la data actual encara faltin 2 o més dies per la data de la reserva. Retorna si es pot anul·lar o no.

### Classe «Auditori» hereta «Espai»

- «capacitat»: aforament de l'auditori, ha de ser un valor superior a 0 (substituir per 1).

`Auditori(edifici, planta, nom, capacitat)`

Constructor inicialitza les dades que arriben des dels paràmetres.

`InfoEquipaments(): String[]`

Retorna un únic text indicant l'aforament de l'auditori:

«L'aforament de l'auditori és de XX persones»

Per veure'n exemples concrets consulteu l'[Annex A](#).

`tipusEspai(): String`

Retorna el text "Auditori".

`anullarReserva(dia): boolean`

No es poden anul·lar les reserves d'un auditori.

### Classe «SalaReunions» hereta «Espai»

- «aacc»: indica si la sala de reunions disposa d'aire condicionat.
- «wifi»: indica si la sala de reunions disposa d'accés WiFi.



`SalaReunions(edifici, planta, nom, aacc, wifi)`

Constructor inicialitza les dades que arriben des dels paràmetres.

`InfoEquipaments(): String[ ]`

Retorna dos texts indicant si disposa dels equipaments:

«Aire condicionat? Si/No  
Accés WiFi? Si/No»

Per veure'n exemples concrets consulteu l'[Annex A](#).

`tipusEspai(): String`

Retorna el text "Sala de Reunions".

### Classe «AulaInformatica» hereta «Espai»

- «equips»: nombre d'equips disponibles a l'aula d'informàtica, ha de ser un valor superior a 0 (substituir per 1).
- «impressora»: indica si l'aula disposa d'impressora.
- «projector»: indica si l'aula disposa de projector.
- «user»: usuari d'accés als equips de l'aula, no pot ser nul (substituir per un text buit).
- «password»: password d'accés als equips de l'aula, no pot ser nul (substituir per un text buit).

`AulaInformatica(edifici, planta, nom, equips, impressora, projector, user, password)`

Constructor inicialitza les dades que arriben des dels paràmetres.

`InfoEquipaments(): String[ ]`

Retorna quatre texts indicant els equipaments i recursos de l'aula:

«L'aula disposa de XX ordinadors  
L'accés als equips és: USER/PWD  
[No] hi ha impressora  
[No] hi ha projector»

Per veure'n exemples concrets consulteu l'[Annex A](#).

`tipusEspai(): String`

Retorna el text "Aula Informàtica".



## Annex A. Exemple sortida programa «MainUF4Pe1»

```
#####
                        ESPAIS
#####

| c/ Joseph Conrad, 111      | Sala de Reunions |
| Telf: 960 55 51 11        | Larry Niven      |
| Mail: contact@nostromo.com | Planta: 1        |
+-----+-----+
|                        EQUIPAMENTS                        |
+-----+-----+
| Aire condicionat? No      |                   |
| Accés WiFi? Si           |                   |
+-----+-----+

| c/ Frank Herbert, 19-65   | Auditori         |
| Telf: 978 55 53 33        | Stanislaw Lem    |
| Mail: atreides@dune.com   | Planta: -1       |
+-----+-----+
|                        EQUIPAMENTS                        |
+-----+-----+
| L'aforament de l'auditori és de 150 persones            |
+-----+-----+

| c/ John Varley, 123-133   | Aula Informàtica |
| Telf: 935 55 53 33        | Poul Anderson    |
| Mail: capitanacirocco@tita... | Planta: -2       |
+-----+-----+
|                        EQUIPAMENTS                        |
+-----+-----+
| L'aula disposa de 15 ordinadors                          |
| L'accés als equips és: guardians/%time1960              |
| No hi ha impressora                                       |
| Hi ha projector                                           |
+-----+-----+

#####
                        ESPAIS + CALENDARIS
#####

| c/ Joseph Conrad, 111      | Sala de Reunions |
| Telf: 960 55 51 11        | Robert Heinlein  |
| Mail: contact@nostromo.com | Planta: 1        |
+-----+-----+
|                        EQUIPAMENTS                        |
+-----+-----+
| Aire condicionat? Si      |                   |
| Accés WiFi? Si           |                   |
+-----+-----+
```

**Tipus d'espai**  
Mètode *tipusEspai()*: String

**Capçalera Sala de Reunions**  
Mètode *capçaleraEspai()*: String

**Equipaments de l'espai**  
Mètode *infoEquipaments()*: String[]



CALENDARI DE RESERVES								
de gener, 2024								
-----								
DL.	DT.	DC.	DJ.	DV.	DS.	DG.		
-----								
XX	2	3	4	5	6	7		
8	9	10	11	12	13	14		
15	16	17	XX	XX	20	21		
22	23	24	25	26	27	28		
XX	30	31	-	-	-	-		
-----								
c/ Ray Bradbury, 451				Auditori				
Telf: 922 55 52 22				Arthur C. Clarke				
Mail: cronicas.marcianas@b...				Planta: 0				
-----								
EQUIPAMENTS								
-----								
L'aforament de l'auditori és de 250 persones								
-----								
CALENDARI DE RESERVES								
de febrer, 2024								
-----								
DL.	DT.	DC.	DJ.	DV.	DS.	DG.		
-----								
-	-	-	1	XX	3	4		
5	6	7	8	9	10	XX		
12	13	14	15	16	17	18		
19	20	21	XX	23	24	25		
26	27	28	29	-	-	-		
-----								
c/ John Varley, 123-133				Aula Informàtica				
Telf: 935 55 53 33				Poul Anderson				
Mail: capitanacirocco@tita...				Planta: -2				
-----								
EQUIPAMENTS								
-----								
L'aula disposa de 15 ordinadors								
L'accés als equips és: guardians/%time1960								
No hi ha impressora								
Hi ha projector								
-----								
CALENDARI DE RESERVES								
d'abril, 2024								
-----								
DL.	DT.	DC.	DJ.	DV.	DS.	DG.		
-----								
1	XX	XX	XX	5	6	7		
8	9	10	11	12	13	14		
15	16	17	18	19	20	21		
XX	23	24	25	26	27	28		



```

| XX | 30 | - | - | - | - | - |
-----
#####
CONSULTES
#####

1) Espai "Poul Anderson" reservat el dia 04/04/2024
2) Espai "Poul Anderson" disponible el dia 05/04/2024

#####
ANUL·LACIONS
#####

3) Anul·lada reserva a "Robert Heinlein" dia 19/01/2024
4) Cap reserva a "Robert Heinlein" el dia 20/01/2024

```

## Annex B. Mètodes disponibles de la classe «Espai»

```

private static final String TITOL_INFORME = "Equipaments";
private static final String TITOL_CALEDARI = "Calendari de reserves";
private static final String BORDER_CHAR = "-";
private static final String MIDDLE_BORDER = " | ";
private static final String LEFT_BORDER = "| ";
private static final String RIGHT_BORDER = " |";
private static final int COL1 = 29;
private static final int COL2 = 17;
private static final int DAY_WIDTH = 3;
private static final int AMPLI_INFORME = 57;
private static final int MIDA_RESERVES = 1000;

/**
 * Dades capçalera de l'espai:
 *
 * Adreça de l'edifici i dades de contacte, tipus d'espai, nom i planta on s'ubica
 * Informació dels equipaments de l'espai
 *
 * @return Text amb la informació de l'espai
 */
public String capçaleraEspai() {
    String ubicacio = "";
    int innerWidth = AMPLI_INFORME - LEFT_BORDER.length() - RIGHT_BORDER.length();

    ubicacio += StringUtils.repeat(BORDER_CHAR, AMPLI_INFORME) + System.lineSeparator();
    ubicacio += LEFT_BORDER + StringUtils.rightPad(StringUtils.abbreviate(edifici.getAdreca(), COL1), COL1);
    ubicacio += MIDDLE_BORDER + StringUtils.leftPad(this.tipusEspai(), COL2) + RIGHT_BORDER + System.lineSeparator();
    ubicacio += LEFT_BORDER + StringUtils.rightPad("Telf: " + edifici.getTelefonFormat(), COL1);
    ubicacio += MIDDLE_BORDER + StringUtils.leftPad(StringUtils.abbreviate(this.nom, COL2), COL2) + RIGHT_BORDER +
    System.lineSeparator();
    ubicacio += LEFT_BORDER + StringUtils.rightPad(StringUtils.abbreviate("Mail: " + edifici.getMailContacte(), COL1), COL1);
    ubicacio += MIDDLE_BORDER + StringUtils.leftPad("Planta: " + this.planta, COL2) + RIGHT_BORDER + System.lineSeparator();
    ubicacio += StringUtils.repeat(BORDER_CHAR, AMPLI_INFORME) + System.lineSeparator();

    ubicacio += LEFT_BORDER + StringUtils.center(StringUtils.upperCase(TITOL_INFORME), innerWidth) + RIGHT_BORDER +
    System.lineSeparator();
    ubicacio += StringUtils.repeat(BORDER_CHAR, AMPLI_INFORME) + System.lineSeparator();

    String[] infoConcrete = this.infoEquipaments();
    if (infoConcrete != null) {
        for (int i = 0; i < infoConcrete.length; i++) {
            ubicacio += LEFT_BORDER + StringUtils.center(StringUtils.abbreviate(infoConcrete[i], innerWidth),
            innerWidth) + RIGHT_BORDER + System.lineSeparator();
        }
    }

    ubicacio += StringUtils.repeat(BORDER_CHAR, AMPLI_INFORME);

    return ubicacio;
}

```



```
/**
 * Graella del calendari de reserves que mostra la informació per a un mes d'un any
 *
 * @return String La graella amb el calendari de reserves
 */
public String calendariEspai(int mes, int any) {
    String calendari = "";
    Locale locale = new Locale("CA", "ES");
    DateFormatSymbols sym = new DateFormatSymbols(locale);
    SimpleDateFormat sdf = new SimpleDateFormat("MMMM, yyyy", locale);
    int innerWidth = AMPLE_INFORME - LEFT_BORDER.length() - RIGHT_BORDER.length();

    Calendar cal = Calendar.getInstance(locale);
    //cal.setFirstDayOfWeek(Calendar.MONDAY);
    cal.set(any, mes - 1, 1);
    mes = cal.get(Calendar.MONTH) + 1; // Evita mesos incorrectes > 12 o < 1
    any = cal.get(Calendar.YEAR);

    // Header
    calendari += LEFT_BORDER + StringUtils.center(StringUtils.upperCase(TITOL_CAENDARI), innerWidth) + RIGHT_BORDER +
    System.lineSeparator();
    calendari += LEFT_BORDER + StringUtils.center(sdf.format(cal.getTime()), innerWidth) + RIGHT_BORDER +
    System.lineSeparator();
    calendari += StringUtils.repeat(BORDER_CHAR, AMPLE_INFORME) + System.lineSeparator();

    // Grid header
    String[] sdays = sym.getShortWeekdays();
    calendari += LEFT_BORDER + StringUtils.upperCase(sdays[Calendar.MONDAY]) + MIDDLE_BORDER;
    calendari += StringUtils.upperCase(sdays[Calendar.TUESDAY]) + MIDDLE_BORDER;
    calendari += StringUtils.upperCase(sdays[Calendar.WEDNESDAY]) + MIDDLE_BORDER;
    calendari += StringUtils.upperCase(sdays[Calendar.THURSDAY]) + MIDDLE_BORDER;
    calendari += StringUtils.upperCase(sdays[Calendar.FRIDAY]) + MIDDLE_BORDER;
    calendari += StringUtils.upperCase(sdays[Calendar.SATURDAY]) + MIDDLE_BORDER;
    calendari += StringUtils.upperCase(sdays[Calendar.SUNDAY]) + RIGHT_BORDER;
    calendari += System.lineSeparator();
    calendari += StringUtils.repeat(BORDER_CHAR, AMPLE_INFORME) + System.lineSeparator();

    // Grid
    int primerDiaMes = Math.floorMod(cal.get(Calendar.DAY_OF_WEEK)-2,7)+1; // Sunday 1 ... Saturday 7 => Monday 1 ....
    Sunday 7

    int diesDelMes = cal.getActualMaximum(Calendar.DAY_OF_MONTH);
    int primeraSetmana = cal.get(Calendar.WEEK_OF_MONTH);
    cal.set(Calendar.DAY_OF_MONTH, diesDelMes); // Últim dia
    int setmanesMes = cal.get(Calendar.WEEK_OF_MONTH) - primeraSetmana + 1;

    int diaMes = 0;
    String[][] monthView = new String[setmanesMes][9];
    for (int i = 0; i < monthView.length; i++) {
        monthView[i][0] = LEFT_BORDER;
        for (int j = 1; j < 8; j++) {
            String sdia = "";
            if ( diaMes == 0 && j < primerDiaMes ) sdia = StringUtils.center(BORDER_CHAR, DAY_WIDTH); // Primera
            if ( diaMes == 0 && j == primerDiaMes ) diaMes = 1;
            if ( diaMes > 0 && diaMes <= diesDelMes ) {
                sdia = StringUtils.center(diaMes+"", DAY_WIDTH);
                diaMes++;
            } else {
                sdia = StringUtils.center(BORDER_CHAR, DAY_WIDTH); // Última setmana
            }
            monthView[i][j] = sdia+ MIDDLE_BORDER;
        }
        monthView[i][8] = System.lineSeparator();
    }

    // Afegir reserves
    for (int i = 0; i < reserves.length; i++) {
        if (reserves[i] != null) {
            cal.setTime(reserves[i]);
            if (cal.get(Calendar.YEAR) == any && cal.get(Calendar.MONTH) + 1 == mes) {
                int diaSetmana = Math.floorMod(cal.get(Calendar.DAY_OF_WEEK)-2,7)+1; // Sunday 1 ...
                Saturday 7 => Monday 1 .... Sunday 7

                int setmanaMes = cal.get(Calendar.WEEK_OF_MONTH) - primeraSetmana;
                monthView[setmanaMes][diaSetmana] = StringUtils.center("XX", DAY_WIDTH)+ MIDDLE_BORDER;
            }
        }
    }

    for (int i = 0; i < monthView.length; i++) {
        for (int j = 0; j < 9; j++) calendari += monthView[i][j];
    }
    calendari += StringUtils.repeat(BORDER_CHAR, AMPLE_INFORME);
    return calendari;
}
```