**R1: Waterloo Station**

Gara Waterloo Station din Londra doreste o miniaplicatie cu ajutorul careia sa vizualizeze informatii despre trenurile care pleaca din gara.

Trenul are ca si atribute importante **destinatia** sa, **ora plecarii** si un **cod** unic.

1. Creati modelul corespunzator pentru gara (clasele gara, tren + atributele sale, folositi **java.util.List**) 2p

2. Creati o clasa separata de test in care sa initializati datele (o gara cu minim 5 trenuri) 2p

3. Afisati la consola informatiile despre trenurile care pleaca din gara iar acestea sa fie sortate ascendent in functie de ora plecarii (folositi **Collections.sort si un Comparator**) 2p

4. Creati o interfata grafica care sa permita urmatoarele:

- introducerea unui cod de tren, apasarea unui buton si afisarea informatiilor intr-un panou 1p

5. Adaugati un ascultator pe buton pentru a afisa detaliile despre trenul cu codul dorit, iar daca trenul nu a fost gasit sa se afiseze un mesaj de eroare corespunzator. 2p

1p oficiu

Exemplu de output in consola pentru sortare la punctul 3:

**Inainte de sortare:------------------------------------------**

**Cod: AC1 Destinatie: Berlin Ora plecare 10:01**

**Cod: AC2 Destinatie: Munchen Ora plecare 05:01**

**Cod: AC3 Destinatie: Frankfurt Ora plecare 16:01**

**Cod: AC4 Destinatie: Dusseldorf Ora plecare 10:31**

**Cod: AC5 Destinatie: Koln Ora plecare 13:01**

**Dupa sortare: -------------------------------------------------**

**Cod: AC2 Destinatie: Munchen Ora plecare 05:01**

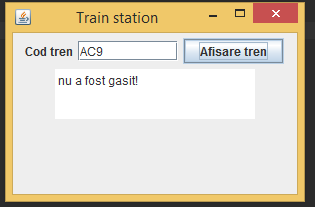
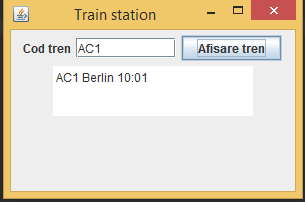
**Cod: AC1 Destinatie: Berlin Ora plecare 10:01**

**Cod: AC4 Destinatie: Dusseldorf Ora plecare 10:31**

**Cod: AC5 Destinatie: Koln Ora plecare 13:01**

**Cod: AC3 Destinatie: Frankfurt Ora plecare 16:01**

Exemplu de interfata grafica la punctul 4 si 5:



**R2: Piraeus Harbour**

Portul Piraeus din Atena doreste o miniaplicatie cu ajutorul careia sa vizualizeze informatii despre vapoarele care pleaca din portul sau.

Vaporul are ca si atribute importante **numarul de locuri**, **ora plecarii** si un **nume** unic.

1. Creati modelul corespunzator pentru port (clasele port, vapor + atributele sale, folositi **java.util.List**) 2p

2. Creati o clasa separata de test in care sa initializati datele (un port cu minim 5 vapoare) 2p

3. Afisati la consola informatiile despre vapoarele care pleaca din port iar acestea sa fie sortate ascendent in functie de ora de plecare (folositi **Collections.sort si un Comparator**) 2p

4. Creati o interfata grafica care sa permita urmatoarele:

- introducerea unui nume de vapor, apasarea unui buton si afisarea informatiilor despre vapor intr-un panou 1p

5. Adaugati un ascultator pe buton pentru a afisa detaliile despre vaporul cu numele dorit, iar daca vaporul nu a fost gasit sa se afiseze un mesaj de eroare corespunzator. 2p

1p oficiu

Exemplu de output in consola pentru sortare la punctul 3:

**Inainte de sortare:------------------------------------------**

**Nume: AC1 Nr locuri: 12 Ora plecare 10:01**

**Nume: AC2 Nr locuri: 14 Ora plecare 05:01**

**Nume: AC3 Nr locuri: 1 Ora plecare 16:01**

**Nume: AC4 Nr locuri: 11 Ora plecare 10:31**

**Nume: AC5 Nr locuri: 22 Ora plecare 13:01**

**Dupa sortare: -------------------------------------------------**

**Nume: AC2 Nr locuri: 14 Ora plecare 05:01**

**Nume: AC1 Nr locuri: 12 Ora plecare 10:01**

**Nume: AC4 Nr locuri: 11 Ora plecare 10:31**

**Nume: AC5 Nr locuri: 22 Ora plecare 13:01**

**Nume: AC3 Nr locuri: 1 Ora plecare 16:01**

Exemplu de interfata grafica la punctul 4 si 5:

