

```

//  

// I. DƏYİŞƏNLƏRİN VƏ ALƏTLƏRİN ELANI (Data Type, Variables)  

//  

/*  

 * TASK 1: Dəyişənləri elan edin (int, double, String, boolean)  

 * ilkinBüdcə: (AZN - Layihənin başlangıç bütçəsi)  

 * komandaPerformansi: (1-100 arası Bal)  

 * LayihəMərhəlesi: (1-ci Mərhələ: Planlaşdırma)  

 * TexnikiBorc: d (Layihə gecikmələri və keyfiyyətsizliyə görə yığılan bərc)  

 * layihəMeneceriAd:  

 * layihəUğursuz:  

 * kritikModulXətası:  

 * Scanner və Random obyektlərini yaradın.  

*/  

//  

=====  

// TƏLƏBƏ A İŞ SAHƏSİ: PLANLAMA VƏ BÜDCƏ NƏZARƏTİ  

//  

=====  

// 1. GİRİŞ VƏ İLKİN KOMANDA NƏZARƏTİ (Scanner, Random, if-else)  

/*  

 * TASK 2: Layihə Menecerinin (Tələbə A) adını Scanner istifadə edərək alın.  

 * TASK 3: ilkinBüdcə dəyişənini təsadüfi dəyerlər yəniden təyin edin (Math.random()).  

 * TASK 4: komandaPerformansi dəyerini təsadüfi int dəyeri ilə yəniden təyin edin (Random sınıfı).  

 * TASK 5: if-else istifadə edərək, komandaPerformansı ??? kiçik olarsa, "Komandada  

Təlim Ehtiyacı Var" xəbərdarlığı çap edin.  

 * TASK 6: "ilkin Təchizat Xərci"ni (ilkinBüdcə * 0.05) təyin edin.  

 * TASK 7: ilkinBüdcə dən ilkin Təchizat Xərci ni çıxın.  

*/  

// 2. LAYİHƏ İCRA MƏRHƏLƏLƏRİ (for Loop)  

System.out.println("\n 2. ERP Tətbiqi Fazaları (5 Modul)");  

int toplamModulSayi = 5;  

/*  

 * TASK 8: 5 dəfə tekrarlanan bir for döngüsü başladın (Modularının icrasını simulyasiya  

edir).  

*/

```

```

-----// Döngü daxilində
-----
/*
 * TASK 9: Mərhələ Xərcini (ilkBüdcə * 0.15 * (1 + (LayihəMərhələsi * 0.05)))
hesablayın.
 * TASK 10: ilkBüdcə dəyişənindən hesablanmış Mərhələ Xərcini çıxın.
 * TASK 11: LayihəMərhələsi dəyərini hər dəfə 1 vahid artırın.
 * TASK 12: komandaPerformansı dəyərini təsadüfi azaldın (Normal Yorğunluq).
 * TASK 13: Random istifadə edərək 20% şansla Texniki Problem yaradın. Baş verərsə
TexnikiBorc-u ??? artırın.
 * TASK 14: ilkBüdcə < 100000.0 olarsa, layihəUğursuz = ??? edərək döngünü
dayandırın (break).
*/
-----/* 
 * TASK 15: TexnikiBorc > 20000.0 olarsa, komandaPerformansi ni 10 vahid azaldın.
*/
-----// 
=====
// TƏLƏBƏ B İŞ SAHƏSİ: KEYFİYYƏT NƏZARƏTİ VƏ RİSK İDARƏETMƏSİ
//
=====

// 3. SİSTEM TEST VƏ KEYFİYYƏT YOXLANMASI (do-while Loop, Scanner)
if (!layihəUğursuz) {
    System.out.println("\n--- 3. Son İstifadəçi Testləri (UAT) ---");

    /*
     * TASK 16: do-while döngüsünü başladın. Döngü komandaPerformansi < 80 VƏ
     TexnikiBorc > 0 olduğu müddətcə davam etməlidir.
    */
-----// Döngü daxilində
/*
 * TASK 17: Scanner ilə istifadəcidən "Düzəliş Priyortetini" seçməsini isteyin (1: Borcu
Sil, 2: Komandanı Stimullaşdır, 3: Gecikməni Qəbul Et).
 * TASK 18: LayihəMərhələsi dəyərini 1 vahid artırın.
 * TASK 19: TexnikiBorc-u 5000.0 azaldın VƏ ilkBüdcə-dən 5000.0 çıxın.
 * TASK 20: komandaPerformansi-nı 8 vahid artırın VƏ ilkBüdcə-dən 8000.0 çıxın.
 * TASK 21: TexnikiBorc-u 2000.0 azaldın VƏ kritikModulXətası-nı 1 vahid artırın.
 * TASK 22: if istifadə edərək, komandaPerformansi > 100 olarsa,
komandaPerformansi = 100 edərək döngünü dayandırın (break).
*/
}
}

```

```

// 4. KRİTİK GECİKMƏLƏR VƏ QƏRRƏLƏR (switch-case, if-else)
if (!layihəUğursuz) {
    System.out.println("\n--- 4. Mühüm Səhvlərə Reaksiya ---");

    /*
     * TASK 23: Scanner ilə xəta tipini istəyin (1: Maliyyə Modulu Xətası, 2: Məlumat İtki Riski, 3: Yeni Tələb).
     * TASK 24: İstifadəçinin seçimini yoxlamaq üçün switch-case blokunu tətbiq edin.
     */

    // switch daxilində
    /*
     * TASK 25: Case 1 (Maliyyə Xətası): if-else ilə ilkinBüdcə < 500000.0-ı yoxlayın. Əgər doğrudursa, layihəUğursuz = true. Əks halda TexnikiBorc-u 15000.0 artırın.
     * TASK 26: Case 2 (Məlumat İtki Riski): komandaPerformansi-nı 10 vahid azaldın VƏ TexnikiBorc-u 10000.0 artırın.
     * TASK 27: Case 3 (Yeni Tələb): Random istifadə edərək 50% şansla ilkinBüdcə-yə 50000.0 əlavə edin (Əlavə Maliyyə Ayırması).
     */
}

// =====
// TƏLƏBƏ C İŞ SAHƏSİ: YEKUN KEÇİD VƏ UĞURUN ÖLÇÜLMƏSİ
// =====

```

```

// 5. KEÇİD VƏ SON İSTİFADƏÇİ TƏMİNATI (TASK 28-37)
System.out.println("\n--- 5. Sistem Keçidinin Nəticələri ---");

/*
 * TASK 28: int tipində "Son İstifadəçi Təminatı Balı"nı (50-90) Random ilə təyin edin.
 * TASK 29: double tipində "Potensial Dəyər Qazancı"nı (Son İstifadəçi Təminatı Balı * 5000.0) hesablayın.
 * TASK 30: ilkinBüdcə-yə Potensial Dəyər Qazancı-nı əlavə edin.
 * TASK 31: 4 dəfə təkrarlanan bir for döngüsü başladın (Keçid sonrası 4 Həftə).
 * TASK 32: Hər iterasiyada komandaPerformansi-nı 1 vahid artırın (Öyrənmə Effekti).
 * TASK 33: Hər iterasiyada TexnikiBorc-u 2000.0 azaldın.
 * TASK 34: LayihəMərhələsi dəyərini 1 vahid artırın.
 * TASK 35: if istifadə edərək, TexnikiBorc < 0 olarsa, TexnikiBorc = 0 edin.
 * TASK 36: Random istifadə edərək 10% şansla kritikModulXətası-nı 1 vahid artırın (Post-Keçid Xətası).
 * TASK 37: LayihəMərhələsi-nin yekun dəyərini çap edin.

```

```
*/
```

## // 6. RİSK PROQNOZU VƏ DƏRSLƏRİN ALINMASI (TASK 38-47)

```
System.out.println("\n--- 6. Gələcək Risk Səviyyəsinin Proqnozu ---");
```

```
/*
```

\* TASK 38: Scanner ilə layihə menecerindən "Alınan Dərslər Balını" (1-10) daxil etməsini istəyin.

\* TASK 39: if-else if istifadə edərək:

\* TASK 40: Bal 8-dən böyük olarsa: komandaPerformansi-nı 5 vahid artırın.

\* TASK 41: Bal 5-dən kiçik olarsa: TexnikiBorc-u 10000.0 artırın VƏ LayihəMərhələsi-ni 1 vahid azaldın.

\* TASK 42: do-while döngüsü başladın. Döngü kritikModulXətası > 0 olduğu müddətce davam etsin.

\* TASK 43: Döngü daxilində kritikModulXətası-nı 1 vahid azaldın.

\* TASK 44: Hər azalmada ilkinBüdcə-dən 2000.0 çıxın (Xəta Təmir Xərcləri).

\* TASK 45: Döngünü dayandırın.

\* TASK 46: double tipində "Layihənin Nəhayı Dəyər İndeksi"ni (Potensial Dəyər Qazancı / 1000 - TexnikiBorc / 1000) hesablayın.

\* TASK 47: Nəhayı Dəyər İndeksini çap edin.

```
*/
```

## // 7. YEKUN NƏTİCƏ VƏ HESABAT (TASK 48-60)

```
System.out.println("\n--- 7. Yekun Layihə Protokolu ---");
```

```
/*
```

\* TASK 48: int tipində "ERP Təsdiq Balı"nı (LayihəMərhələsi \* 10 + komandaPerformansi) hesablayın.

\* TASK 49: Scanner ilə "Yekun Təsdiq Kodu"nu (int) daxil etməsini istəyin.

\* TASK 50: switch-case istifadə edərək kodu yoxlayın (1: Təsdiq, 2: Audit, 3: Qadağa).

\* TASK 51: Case 1 (Təsdiq): ilkinBüdcə-yə 200000.0 əlavə edin (Bonus).

\* TASK 52: Case 2 (Audit): TexnikiBorc-u 50000.0 artırın VƏ layihəUğursuz = false edin.

\* TASK 53: Case 3 (Qadağa): layihəUğursuz = true edin VƏ ilkinBüdcə-dən 50000.0 çıxın.

```
*/
```

```
/*
```

\* TASK 54: if-else if-else istifadə edərək 3 fərqli sonluq yazın:

\* - 1) Mükəmməl Uğur (LayihəMərhələsi > 8 VƏ TexnikiBorc == 0).

\* - 2) Şərti Keçid (layihəUğursuz == false VƏ ilkinBüdcə > 100000.0).

\* - 3) Layihə Fəlakəti (layihəUğursuz == true VƏ kritikModulXətası > 0).

```
*/
```

```
/*
```

\* TASK 55: double tipində "Gələcək Xidmət Xərci"ni (TexnikiBorc \* 0.5 + 50000.0) hesablayın.

\* TASK 56: Yekun LayihəMərhələsi dəyərini çap edin.

\* TASK 57: Yekun TexnikiBorc dəyərini çap edin.

\* TASK 58: Yekun ilkinBüdcə dəyərini çap edin.

\* TASK 59: Yekun komandaPerformansı dəyərini çap edin.

\* TASK 60: Scanner-i bağlayın.

\*/