

```

/* -----
*
*           PALACSINTA ELKÉSZÍTÉSE
*           mérnök informatikusoknak és programozóknak
*
*
*   Kiötlötték, majd billentyűzettel digitalizálták:
*   =====
*   Sicz-Mesziár János (http://szyrt.hu/)
*   Molnár Sándor (http://aleeex.extra.hu/)
*
*   C# szintaktika megismeréshez kiváló. Fogyasszátok egészségel!
* -----* */

```

```

using System;
using System.Collections;
using Konyha.Eszközök; //tálak, kanalak
using Konyha.DefaultAnyagok; //tojás, tej, töltelékek, stb...
using Konyha.Skill; //késztségek, pl tojástörés

namespace Szakácskönyv
{
    class Hogyan_készül_a_palacsinta{
        static void Main(string[] args){
            try
            {
                // 1.) Adjuk hozzá az alapanyagokat
                Palacsinta palacsintakirály = new Palacsinta(
                    SÓ.csipet, 25, CUKOR.evőkanál * 1,
                    0.3, 0.2, 3,
                    0.2
                );
                // 2.) Töltelékek meghatározása
                palacsintakirály.AddTöltelék(new string[]{"", "kakaó", "lekvár", "túró", "fahéj"});
                // 3.) Készítsük el
                palacsintakirály.Start();
                // 4.) Jó étvágyat
                System.Console.WriteLine("Jó étvágyat");
            }
            catch (Exception ex)
            {
                getProblem();
                Process.Start("http://www.google.hu/search?q=palacsinta+rendelés");
            }
        }
    }

    // Elkészítés

```

```

public class Palacsinta
{
    // Alapanyagok
    private int só, liszt, cukor; //dkg
    private int tej, olaj, ásványvíz; //liter
    private int tojás; //darab
    private List<KészPalacsinta> palacsinták = new List<KészPalacsinta>();
    private List<Töltelékek> palacsintaTöltelékek = new List<Töltelékek>();

    //Eszközök a konyhából
    private Tál egyTál = new Tál();
    private Kanál keverőKánál = new Kanál();
    private Serpenyő egySerpenyő = new Serpenyő();
    private Spakli forgatóSpakli = new Spakli();

    // probléma elhárítás, véletlen
    private string last_problem;
    private Random rnd = new Random();

    public Palacsinta(int só, int liszt, int cukor,
        int tej, int olaj, int tojás,
        int ásványvíz)
    {
        // Ennyi kell, se több, sem kevesebb
        this.só = só;
        this.liszt = liszt;
        this.cukor = cukor;
        this.tej = tej;
        this.olaj = olaj;
        this.tojás = tojás;
        this.ásványvíz = ásványvíz; // lehet null
        this.last_problem = null;
    }

    public void AddTöltelék(string[] töltelékek)
    {
        foreach(string t in töltelékek)
            this.palacsintaTöltelékek.Add(t);
    }

    public void Start()
    {
        if(Kever()){
            System.Threading.Thread.Sleep(30 * 60 * 1000);
            int count = 0;
            foreach(PalacsintaTészta adag in this.egyTál.weighOutAs<PalacsintaTészta>())
                if (Sütés(adag)) count++;
        }
    }
}

```

```

        for (int i = 1; i <= count; i++)
            palacsinták[i].Töltés(this.palacsintaTöltelék[this.rnd.Next(1,5)]);

    }else{
        getProblem();
    }
}

public bool Kever()
{
    try
    {
        this.egyTál.Kanál = this.keverőKanál;
        this.egyTál.Add(loadTej(this.tej)); //tej
        for (int i = 0; i < this.tojás; i++)
            this.egyTál.Add(breakTojás(TojásKészlet[i]).getAll());
        this.egyTál.Add(this.só);
        this.egyTál.Add(this.cukor);
        this.egyTál.Mix();
        if (this.ásványvíz != null)
            this.egyTál.Add(loadVíz(this.ásványvíz));
        this.egyTál.Mix();
        this.egyTál.Add(loadLiszt(Liszt));
        while(this.egyTál.CurrentItem.State != Csomómentes)
            this.egyTál.Mix();
        return true;
    }
    catch (OverflowException oe) // több a cucc, mint amekkora a tál
    {
        this.last_problem = oe.Message;
    }
    catch (Exception ex) {
        throw new Exception(ex.Message);
    }
    return false;
}

public bool Sütés(PalacsintaTészta adag)
{
    if (adag != null)
    {
        this.egySerpenyő.Clear();
        this.egySerpenyő.Add(loadOlaj(this.olaj));
        if (this.egySerpenyő.Hot)
            this.egySerpenyő.Add(adag);
        System.Threading.Thread.Sleep(2 * 60 * 1000); //egyik oldal
    }
}

```

```
        if (Serpenyő.Rotate(this.forgatóSpakli))
            System.Threading.Thread.Sleep(1 * 60 * 1000); //másik oldal
        if (!Carbon.equals(this.egySerpenyő.CurrentItem))
        {
            palacsinták.Add((KészPalacsinta) this.egySerpenyő.CurrentItem);
            return true;
        }
    }
    else
    {
        throw new Exception("Nincs adag");
    }
    return false;
}

public void getProblem()
{
    if(this.last_problem != null)
        System.Console.WriteLine(this.last_problem);
}
}
```