Sąsaja – abstraktus objekto klasės aprašas, nusakantis, kokius metodus ši klasė privalo turėti.

Realizacijos aprašas, paveldi daug, bet kuri klasė. Virtualūs metodai, nebūtinai realiz, glaudi sąsaja – viena klasė.

Įgyvendinta sistemoje, per paveldėjimą, klasė įsipareigoja 2 dalykai.

Aprašymas: keletas metodų, tiksliai sutampa metodų vardai ir parametrai, atviri metodai – jokių prieigos modifikatorių.

Iteratoriai - tai apibendrintos rodyklės, jungiančios konteinerių elementus. Didinant ar mažinant iteratoriaus reikšmę, pasiekiamas vis kitas konteinerio elementas.

Objekto sukūrimas/naikinimas. inic

Destruktorius – specialus objekto metodas, atsakingas už visų objekto užimamų resursų (atminties, užmegztų tinklo jungčių, laikinų bylų) užleidimą kitiems tikslams. Baigus vykdyti destruktoriaus kodą, objektas laikomas „sunaikintu“ – jokie vėlesni kreipiniai į jį nebeleistini.

Šiukšlių surinkėjas (GC, garbage collector) – vykdomosios sistemos dalis, automatiškai naikinanti vykdomojo kodo nebepasiekiamus objektus tam, kad nesibaigtų atmintis.

Ką užtikrina? Destruktoriai, tik kartą, neaktyvūs

Objektų kartos. Nauji, išgyvenę valymą, nes užteko atminties, išgyvenę kelis.

Resursų grąžinimo sąsaja. Automatiniam apdorojimui, pvz, using

Delegatas – rodyklė į metodą ar jų sąrašą. Iškvietus delegatą, vykdomas tas metodas, į kurį jis rodo. Yapč naudingi, kai yra keletas objektų, turinčių vienodas metodų antraštes, bet skirtingus metodų vardus.

Poreikis. Kviesti, siųsti kviečiant ir gauti, pertraukti-grįžti tęsti.

Įvykis – klasė, stebinti programos dalį ir aptinkanti pasikeitimą, kažką svarbaus ar netikėto (pelės mygtuko, klavišo paspaudimą; programos pranešimą). Įvykis turi metodų, prenumeratorių, kuriuos galima iškviesti dėl įvykio, sąrašą.

Statiniai metodai – algoritmai, priklausomi tik nuo konkrečios klasės statinių duomenų ir bendri visiems tos klasės objektams. Jie gali būti iškviečiami net ir nesukūrus nei vieno tos klasės objekto, tačiau norint, kad metodas galėtų naudotis klasės aprašymu, būtina tai padaryti. Tokie metodai taip pat dažnai naudojami objektų kūrimui, radimui, naikinimui.

Nuoseklinimas (serialization) – procesas, kai objektų būsenos ar duomenų struktūros išsaugomos laikmenoje tam, kad jas būtų galima pernešti tarp atskirų programų.

Reikalavimai. Atributas [Serializable, sąsaja ISer.

Formatai. Dvejetainis, XML, JSON.

Bendriniai del. Daug parametrų, Action – metodams, kurie negrąžina reikšmės, Func.

Anoniminiai metodai, kurie neturi vardo, bet gali turėti parametrų sąrašą. Iš karto aprašomi toje vietoje, kur būtų kviečiamas įprastas metodas. Jie reikalingi delegatams, nenorint kurti atskiro metodo.

A ir lokalūs kintamieji. Gali pasiekti metodo ir klasės, negali ref out parametrų. Vardai sutapti su klasės.

Išimtis – neįprastas ar išskirtinis programos vykdymo atvejis, kuriam reikalingas specialus apdorojimas.

Pertraukimų (išimčių) apdorojimas (exception handling) – programavimo kalbų mechanizmas, skirtas aprašyti programos reakciją į vykdymo klaidų atsiradimą ir kitas galimas problemas (vadinamąsias išimtis).

Sisteminės išimtys. Jų labai daug, nes vartotojo patogumui jau aprašyti visi galimi atvejai. Jais pasinaudoti galima per SystemException() klasę.

Bendrosios išimtys.

Savybės. Message, HelpLink, TargetSite, Source.

Try: vykdydama šį bloką, programa aptinka išimtis. Kai pasirodo išimtis, vykdymas nutraukiamas ir einama į catch bloką.

Catch: visada rašomas po try. Jame esantys sakiniai nusako, kas turi būti atliekama esant išimčiai. Šių blokų turi būti tiek, kiek skirtingų tipų išimčių apibrėžta throw sakiniuose.

Throw: sakinys, kurį vykdant generuojama išimtis ir pereinama į išimties apdorojimo bloką.

Finally: blokas, kuris būtinai bus įvykdytas, nesvarbu, ar try blokas nutrauktas su komanda break, ar return. Šis blokas užtikrina, kad net įvykus klaidai svarbus programinis algoritmas bus įvykdytas.

Vartotojo išimčių klasė: paveldėti ApplicationException, vardas baigtis Exception, 3 konstruktoriai.

Išimtys gali būti vykdomos ir objekto viešuosiuose (get/set – skaitymo/keitimo) metoduose, kurie reikalingi manipuliuoti objekto vidiniais duomenimis. Šitaip galima kontroliuoti, kokios reikšmės suteikiamos objekto elementams.

Reikšmių kintamieji: saugomi dėkle; vardas susietas su atminties vieta, saugančia reikšmę; kitas kintamasis, gavęs esamo kintamojo reikšmę, saugomas naujoje vietoje, išskyrus parametrus su ref ir out.

Nuorodų: reikšmė – adresas į objekto egzempliorių, saugantį duomenis; iš kintamojo perimamas adresas, pagal adresą išrenkami duomenys; atmintis tiesiogiai susieta su kintamuoju.

Bendrinės klasės (mazgas) – inkapsuliuoja operacijas, kurios nepriklauso konkrečiam duomenų tipui (susietuosius sąrašus, eiles).

Tipai. Tekstiniai/simboliniai. Skaitiniai/aritmetiniai. Loginis.

Struktūra – konstrukcija, skirta sudėtingų duomenų tipų kūrimui

Skirtumai nuo klasės: negalimas paveldėjimas, reikšmių/nuorodų tipai, negali turėti konstr be parametrų ir visiems laukams reikia priskirti reikšmes, jokių virtualių narių.

T?: pasiskolina operatorius iš T, kiti operatoriai grąžina null, jei bent vienas operandas null. ?? kairiojo operando reikšmė grąžinama, jei ne null.

Pritaikymai: duom bazės, neužpildytos reikšmės; pranešimas apie nesėkmę, nenaudojant loginio kint, išvestinėse klasėse.

Kai gaunami netinkami rezultatai, reikia diagnostinės informacijos. Tam sudaromi kodo kontraktai. Jie leidžia metodams bendrauti per tarpusavio įsipareigojimus. Pažeidus šiuos kontraktus, metodai išduoda diagnostiką.

Sąlyginis kompiliavimas. Kompiliatoriaus direktyvomis #if, #else ar #endif galima apibrėžti metodus, kurie vykdomi nusakius direktyvą #define ir kompiliavimo raktą (šiuo atveju TESTMODE), rašomą didžiosiomis raidėmis, failo viršuje. &&, ||, undef

Privalumai: galima įjungti atributą, keisti deklaruojamą tipą.

Atributas Conditional. Nurodo kompiliatoriui ignoruoti visus metodo kvietimus, jei neapibrėžtas nurodytas simbolis (#undef SIMBOLIS). Priešingu atveju (#define SIMBOLIS) visi metodo kvietimai vykdomi.

Kodo kontraktai. Seka programos klaidas, kai programa paleista, ir užtikrina, kad visi duomenys įeinantys ir paliekantys metodus, yra teisingi. Visą informaciją galima įrašyti į dokumentacijos failus ar sekti vykdymo metu.

Listeners pėdsakų klausytojų kolekcija. Write pranešimai į derintojo langą, Fail dialogo langas, Trace. Tai metodai

Derinimo klasės. Statinės, suteikia įrašymo-teigimo galimybes, debug derinimui, trace ir eksploatavimui. Apibrėžti su [conditional, abi Write, Fail, Listeners.