

Модульдік тестілеу(Unit testing)

- **Модульдік тестілеу**-бұл жеке алынған модульдер, функциялар немесе кластар деңгейінде бағдарламаны тестілеу.
- *Модульдік тестілеудің* мақсаты модульде локализацияланған алгоритмдерді іске асырудағы қателерді анықтаудан, сондай-ақ жүйенің келесі әзірлеу мен тестілеудің деңгейіне көшуге дайындық дәрежесін анықтаудан тұрады.
- *Модульдік тестілеу* "ақ жәшік«принципі бойынша жүргізіледі.
- *Модульдік тестілеу* әдетте әрбір модульдің айналасында белгілі бір ортаны құруды білдіреді.

Тест құру принциптері

- **Басқару ағынын талдау негізі.** Бұл жағдайда тесттен өту кезінде жабылатын элементтер C0, C1, C2 тестілеудің құрылымдық өлшемдері негізінде анықталады. Оларға бағдарламаның (МБЖ) бағанын басқарушының шыңдары, доғалары, жолдары, шарттары, шарттардың комбинациясы және т. б. жатады.
бермейді.

Тест құру принциптері

- *Танымал критерийлерге бағдарлама функцияларын жабу критерийі* (бағдарламаның әрбір функциясы кемінде бір рет шақырылуы тиіс) және шақыруларды жабу критерийі (бағдарламада әрбір функцияның әрбір шақыруы кемінде бір рет жүзеге асырылуы тиіс) жатады.

- **Деректер ағынын** талдау негізінде (жабылуға тиіс элементтер бағдарламаның ақпараттық бағаны негізінде анықталады). Бұл түрі айнымалыны пайдалану мен айқындауды қамтитын барлық өзара байланыстардың тестілеуін неинициализацияланбаған айнымалылар мен артық тағайындау сілтемелерін анықтауға бағытталған. Стратегияның кемшілігі, ол шешімдерді жабуға көпшілік

Тест жиынтығын құру

1. МБК құрастыру
2. Тест жолдарын таңдау:
 - Статикалық әдістер
 - Динамикалық әдістер
 - Іске асырылатын жолдардың әдістері
3. Тест жолдарына сәйкес тестерді генерациялау

Көптеген тесттерді құру әдістері

- **Статикалық әдістер.** Шығыс шыңына жеткенше доға қосу арқылы оны біртіндеп ұзарту арқылы әрбір жолды құру. Кемшіліктер-салынған тестілеу жолдарының мүмкін іске асырылмауы ескерілмейді(ақаудың болжанбаған пайызы). - Еңбек сыйымдылығы (жабатын көптеген жолдардан толық тест жүйесіне өту қолмен жүзеге асырылады)
Достоинство - сравнительно небольшое количество необходимых ресурсов

- **Динамикалық әдістер.** Берілген критерияны қанағаттандыратын тестілердің толық жүйесін құру, жапқыш көптеген жолдар мен тестілік деректерді құру есебін бір мезгілде шешу жолымен. Бұл ретте бұрын қаралған жолдардың немесе олардың бөліктерінің іске асырылуын немесе іске асырылмауын автоматты түрде ескеруге болады. Абырой - кейбір сапалы деңгей-жолдардың іске асырылуы.
- **Іске асырылатын жолдардың әдістері.** Көптеген жолдар жиынынан көптеген жолдарды жабатын барлық іске асырылатын жолдарды бөліп алу.

Әдістерді салыстыру

- **Статикалық әдістердің** артықшылығы қажетті ресурстардың салыстырмалы аз мөлшерін құрайды. Алайда оларды іске асыру некенің болжанбаған пайызын (іске асырылмайтын жолдарды) қамтуы мүмкін. Сонымен қатар, бұл жүйелерде жапқыш көптеген жолдан толық тест жүйесіне ауысуды пайдаланушы Қолмен (көп еңбекті қажетсінетін) жүзеге асыруы тиіс.

- **Динамикалық әдістер** әзірлеу кезінде де, пайдалану кезінде де едәуір үлкен ресурстарды талап етеді, алайда шығындардың ұлғаюы негізінен жолдың іске асырылуын анықтау аппаратын әзірлеу және пайдалану есебінен (символдық интерпретатор, теңсіздіктің шешуші) болады. Бұл әдістердің артықшылығы-олардың өнімдерінің сапасы біршама жоғары-жолдардың іске асырылуы. Іске асырылатын жолдардың әдістері ең жақсы нәтиже береді.

Жөндеу әдістері

- Тестті орындау нәтижесі қате болған жерде ештеңе айтпайды. Қатені түзету процедурасы келесі әдістердің көмегімен аралық есептеу хаттамасын талдау болып табылады:
- "Выполнение программы в уме" (deskchecking).
 - Аралық нәтижелерді (logging) хаттамалау операторларын енгізу.
 - Бағдарламаны қадамдық орындау.
 - Тапсырыс берілген аялдамалармен (breakpoints), трасса (traces) немесе жады жағдайын талдау - дамбаларды (dump) орындау. кері орындау (reversible execution) – барлық трассаны орындау тарихының log файлында бекітілген кезде off-line талдау режимінде мүмкін.