Дәріс 11

Объектіге бағытталған талдау мен жобалау негіздері. Объектіге бағытталған талдау мен жобалауға кіріспе.

Лекция жоспары:

1. Объектілік ұстанымға негізделген программалық жабдықтардың ерекшеліктері және оларға қойылатын талаптар
2. Программалық жабдықтарды тестілеу. Программалық жабдықтардың сапасын тексерудің түрлері

Лекция мазмұны

1. Объектілік ұстанымға негізделген программалық жабдықтардың ерекшеліктері және оларға қойылатын талаптар

Объектіге бағдарланған программалау – бұл программалық жабдықты, қандай да болмасын кластың өкілі болып табылатын, объектілердің жиынтығы түрінде құратын программалау методологиясы.

Объектіге бағдарланған жобалау – бұл құрылатын ақпараттық жүйенің (немесе

программалық жабдықтың) барлық статикалық және динамикалық модельдерін объектілі декомпозициялау процесі мен модельдердің логикалық, физикалық тұрғыдан беру тәсілдері негізінде жобалау методологиясы.

Объектіге бағдарланған талдау – бұл жобаланатын жүйеге қойылатын талаптар, пәдік облыстағы анықталған кластар мен объектілер тұрғысынан қарастырылатын методология. Объектіге бағдарланған ұстанымның концептуалдық негіздеріне объектіге бағдарланған ұстанымның моделі жатады. Объектілік модельдеудің негізгі элементтері: *абстаркциялау, инкапсуляция, модульділік* және *иерархия*. Қосымша элементтері*: типтелу, паралеллизм* және *тұрақтылық.*

Абстракциялау – бұл қандай да болмасын объектіні, өзге объектілерден ажырататын белгілері, сипаттамалары және т.б. арқылы бөліп алу, жалпы абстракциялау объектінің сыртқы ерекшеліктеріне негізделеді. Объектіге бағдарланған ұстанымда, берілген объектінің дұрыс абстракциялануы, жобалаудың негізгі міндеттерінің бірі болып саналады.

Инкапсуляция – бұл объектінің, өзінің ішкі элементерінің, бір бірінен ажыратылу процесі. Бұл процесс кезінде объектінің ішкі құрылымдары мен оқиғалары бір- бірінен дұрыс ажыратылады.

Инкапсуляция объектінің интерфейсін қорғау үшін қолданылады немесе объектілік ұстанымда класстың ресурстарын, тек оның өзінің ғана пайдалануын қолдайды. Абстракциялау мен инкапсуляция бірін бірі толықтырады.

Модульділік – бұл программалық жабдықтың декомпозициялану

кезінде өзара байланысқан, бірақ өте әлсіз байланысқан модульдерге бөліну қасиеттері. Инкапсуляция мен модульділік қасиеттері абстракцияларды бір- бірінен ажыратады.

Иерархия – бұл жүйедегі абстракцияланудың бір- біріне бағынышты түрде реттеліп орналасуын тағайындайды. Бұл күрделі жүйедегі класстардың құрылымы (иерархиясы). Мысалы, жай және көп қабылдаушылықты айтуға болады.

Типтелу – бұл абстракцияға байланысты класстарды бір бірінен ажырату үшін қойылатын шектеулер. Паралеллизм – бұл объектінің актив және пассив түрде болуын көрсетеді.

Тұрақтылық – бұл объектінің өмір сүру уақытын көрсетеді. Объектіге бағдарланған ұстанымның негізгі түсініктері: объект, класс. Объект класстың экземпляры тұрғысынан қарастырылады. Объектінің күйі, оқиғасы және жеке қасиеттері болады. Объектіге әсер етуді әдіс деп атайды. Класс қабылдаушылық пен инкапсуляция және полиморфизмді (абстракцияны) қанағаттандыратын құрылымдық жиынтық тип ретінде қабылданған.. Объектіге бағдарланған анализ бен жобалау әдістері модельдеу тілі мен модельдеу

процестерінің сипаттамаларынан тұрады. Модельдеу тілі жобаның сипаттамасын беру үшін қолданылатын нотация.

Нотация – бұл модельдерде қолданылатын графиктік объектілердің жиынтығы. Модельдеу тілінің синтаксисі де нотациямен анықталады.

Процесс – бұл жобаны құру кезінде жасалатын қадамдардың сипаттамалары. UML (Unified Modeling Language) – бұл 1980-1990 ж. қолданылып келген, объектіге бағдарланған анализ бен жобалаудың орнына келген әдіс болып табылады. UML алу үшін бірнеше авторлардың әдістерін біріктіруге тура келді: Boosh – авторы Гради Буч; OMT (object modeling technique) – авторы Джеймс Рамбо; OOSE (object oriented SoftWare engineering) – авторы Ивар Якобсон. ***UML тілінің негізгі мақсаттары мен мүмкіндіктері***:

- қолданушыға түсінікті болатын визуальды модельді құру;
- модельдегі базалық концепциялардың кеңейтуге бейім болуы;
- программалау тілдеріне, құру процессіне тәуелсіз болуы;
- модельдеу тілінің формальды негізде болуын қамтамасыз етеді;
- объектілік бағдарланған жабдықтар нарығына стимуляция жасайды;
- практикалық тәжірибелердің ең жақсысын біріктіру және тарату;

2. Программалық жабдықтарды тестілеу. Программалық жабдықтардың сапасын тексерудің түрлері

Программалық жабдықтың өмірлік циклының бір кезеңін програмалық жабдықты тестілеу процесі құрайды. Программалық жабдықтарды тестілеу өте көп уақытты қажет ететін күрделі де ұзақ процесс. Программалық жабдықты тестілеудің мақсаты бұл құрылған программаның бастапқы техникалық тапсырмаға сәйкес толық рындалуын тексеру және оны қолданысқа енгізгенге дейін мүмкін болатын қателерін табу болып есептеледі. Тестілеу процесі программалық жабдыққа қатысты жасалатын валидация және верификация процесстерінің құрамына кіреді. Халықаралық ISO 9000:2000 стандарты бойынша, валидация (validation)- программалық жабдықтың қолданушының немесе тапсырыс берушінің нақты талаптарын дәл және толық қанағаттандыратындығын объективті фактілер негізінде дәлелдеу үшін жүргізілетін процесс. Верификация (verification)- программалық жабдықтың сапасына

қатысты ішкі ережелер мен стандарттарға спецификациялардың сақталған-сақталмағандығына тексеру үшін жасалады. Мысалы, программалық жабдықты орындауға жіберіп, белгілі бір мәндер үшін шыққан нәтиженің дұрыс-бұрыстығын тексеру валидация процесіне жатады, ал программаны орындауға жібермей-ақ, оның кодын рецензиялау, синтаксистік жазылуларын тексеру және т.б. верификациялау процесіне кіреді. Қазіргі уақытта программалық жабдықтардың сапасын тексеру үшін жүргізілетін тестілеу процесстерін классификациялау бірнеше категориялар бойынша жасалған. Программалық жабдықтың өзін тестілеу объектісі ретінде қарастыратын **тестілеу процесстері**не келесілер жатады:

- функционалдық тестілеу (functional testing);
- өнімділікке тестілеу (performance testing);
- жүктемелерге тестілеу (load testing);
- стресс-тесілеу (stress testing);
- тұрақтылыққа тестілеу (stability / endurance / soak testing);
- юзабилити-тестілеу (usability testing);
- қолданушының интерфейсін тестілеу (UI testing);
- қауіпсіздікке тестілеу (security testing); - локализацияға тестілеу (localization testing); - үйлесімділікке тестілеу (compatibility testing).

Тестілеу процесінің автоматтандырылу дәрежесіне байланысты келесі топтарға бөлінеді:

- қолмен тестілеу (manual testing);
- автоматтандырылған тестілеу (automated testing) ;
- жартылай автоматтандырылған тестілеу (semiautomated testing). Тестілеу процесінде қолданылатын **теориялық әдіс- тәсілдер**ге немесе механизмдерге байланысты төмендегідей түрлері анықталған :
- «қара жәшік» әдісі бойынша тестілеу (black box); - «ақ жәшік» әдісі бойынша тестілеу (white box); - «сұры жәшік» (grey box).

Программалық жабдықтың құрамына кіретін компоненттерінің бір- біріне тәуелсіздігін немесе өзара байланыстарын тексеру мақсатында жүргізілетін тестілеу түрлері:

- компоненттік (модульдік) тестілеу (component/unit testing);
- интеграциялық тестілеу (integration testing);
- жүйелік тестілеу (system/end-to-end testing).

Программалық жабдықты уақытқа қатысты алғанда тестілеудің төмендегідей түрлері қолданылады:

- альфа-тестілеу (alpha testing)
- қабылдау кезіндегі тестілеу (smoke testing)
- жаңа қызметтерге тестілеу (new feature testing)
- регресстік тестілеу (regression testing)
- тапсыру кезіндегі тестілеу (acceptance testing)
- бета-тестілеу (beta testing)

Программалық жабдықтардың сапасын көтеру мақсатында жыл өткен сайын тестілеу процестерінің қатары жаңа әдістермен, жабдықтармен толығып келе жатқанын байқауға болады.

Бақылау сұрақтары

1. Объектіге бағытталған талдау?
2. Объектіге бағытталған жобалау?
3. Объектіге бағытталған программалау?

Ұсынылатын әдебиеттер

1. Гамма Э. Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирование. Паттерны проектирования. – СПб: Питер, 2001.
2. Ишкова Э.А. С++ начала программирования. – М.:Бином, 2001.

Кетков Ю., Кетков А. Практика программирования: Visual Basic, C++ Builder, Delphi