2

1

Springдан амалиётда фойдаланиш усуллари

**Таржимондан**

Азиз китобхон агарда сиз ушбу китобни мутолаа қилишни бошлаган бўлсангиз, демак бизларга мазкур китобни ўрганишни насиб қилиб турган яратганга шукроналар айтамиз. Шунингдек мазкур китобни ўзбек тилига таржима қилишни ушбу муборак рамазон ойида ҳамда қолаверса Муҳтарама Волидамниинг туғлиган кунларида бошлашни насиб қилдиргани учун яратганга чексиз ҳамдлар айтаман. Барча ишларга қодир ҳисобланган Аллохнинг ўзи ота-оналаримизнинг умрлариини зиёда қилсин.

Мазкур китоб Willie Wheeler ва Joshua White қаламларига мансуб бўлган «Spring in Practice», яъни «Springдан амалиётда фойдаланиш усуллари» китобидир. Бошкача қилиб айтадиган бўлсак Spring технологияларини амалиётга жорий этиш усуллари ҳақидаги китобдир.

Ушбу китоб орқали Spring технологиясидан фойдаланган ҳолда иловалар ишлаб чиқиш, ҳамда ундан лойиҳаларда самарали фойдаланиш усулларини ўз ичига олган яна битта китобни ўрганиш арафасида турибмиз. Мазку китобдан ўрганувчиларимиз Spring ҳақидаги билимларини шунингдек, ундан иловалар яратишда кандай фойдаланиш зарурлиги ҳақидаги билимларни олишлари мумкин.

Шуни алоҳида таъкидлаш жоизки ушбу китобнинг муаллифлари узоқ йиллар давомида турли йирик ташкилотларда дастурий махсулотлар ишлаб чиқариш соҳасида самарали фаолият олиб борганлар ҳамда иловалар ишлаб чиқиш соҳасида чуқур тажрибага эга ҳисобланишади. Яъни мазкур китобнинг ёзувчилари замонамизнинг етук дастурчи инженерлари ҳисобланишади.

**Муаллифлар муқаддимаси**

Мен фаолиятимнинг дастлабки бир неча йиллари давомида ўзим иштирок этган лойиҳаларда Enterprise Java Beans(EJB)дан кенг фойдаланишимга тўғри келган, кейинчалик эса айнан 2004 йилда Springдан фойдаланишни бошладим. Уша пайтдаги даврларда кўпчиликдан фарқли равишда, менда EJB ҳақида у даражата ташвишга соладиган ҳикоялар бўлмаган. Шунингдек уша пайтлардаги мен иштирок этган EJB лойиҳаларим жиддий технологик муаммолар вужудга келадиган даражада улкан бўлмаган, балки ҳажми кичикрок бўлгани учун ҳам деярли ҳар кандай технологиялар асосида ҳам амалга оширишиш мумкин эди. Шахсан ўзим EJB билан ишлашни унчалик ёқтирмаган бўлсамда, лекин менинг жамом бу ишларни самарали амалша оиширишга мувоффақ бўлди, шу сабали биз ишлаб чиққан лойиҳалардан фойдаланувчилар тамонидан бизга унақа кўп шикоятлар келиб тушмаган.

Орадан бироз муддат ўтиб мен 2004-йилда бошқа бир ташкилотга ишга кирдим ва ўша ерда ўз фаолиятимни давом этдирдим, у ердаги барча дастурчилар Springдан кенг фойдаланар эди. У ердаги фаолиятимнинг илк кунларидан бошлаб мен шуни англаб етдимки, Springнинг POJO ва инекцияга асосидаги боғланишларни EJB билан солиштирганда Spring фойдаланиш учун анчагина содда ҳамда коднинг тозалигини таъминлаши мумкин экан. Кейинчалик бизнинг жамоамиз тамондан Spring ёрдамида ишлаб чиқилган иловаларимиз минглаб сондаги фойдаланувчиларни хеч қандай муаммоларсиз қўллаб-қувватлашини ўз тажрибаларимизда кўрдик. Spring ўша вақтларда мавжуд бўлган анъанавий қарашларга зид тарзда, яъни EJB ва катта ҳажмдаги илова серверларидан фойдаланмаган ҳолда ишлатишни таклиф қиларди. Шу билан бирга Spring корхона ва ташкилотлар бизнес талабларини қаноатлантириши учун етарлича даражадаги кенг имкониятларга эга эди.

Шу тариқа мен ўз жамомиз билан биргаликда бир нечта Spring иловаларини ишлаб чиқишга мувофақ бўлдик. Шу орада камина Менежер сифатида жуда кўп амалий ишланмаларни яратишни тажрибада амалга ошириб кўрдим ва худди шу тариқа мазкур Frameworkни ўрганиб чиқишимга тўғри келган. Орадан бироз муддат ўтгач менинг ташкилотдаги рахбарлик маъсулиятим ортиб бориши ҳисобига, ўзим истганимдек янги билимларни ўрганишим ҳамда ўз устимда ишлашим учун кам қақт қолар эди. Мен эса мазкур Framework ҳақидаги эгаллаган мавжуд билимларимни сақлаб қолиш ва уни бойитиб боришни мақсад қилган ҳолда Spring (springinpractice.com) ҳақида блог юритишни бошладим. Орадан бироз муддат ўтиб Manning нашриёти менинг блогимдаги маълумотлар билан танишиб чиқиб, менга мазкур китобни ёзишни сўрашди. Мана шундай бир ҳолатда мазкур китобни ёзиш гояси вужудга келди. Ҳозирги кунга келиб мен яна амалий ишланмалар устида тинимсиз ишламоқдаман, шунингдек бу кунгача мен деярли барча Java ишланмаларимда Springдан фойдаланиб келмоқдаман. Бир сўз билан айтганда Spring бу иловаларни ишлаб чиқиш учун жуда ҳам ёқимли ва самарали ҳисобланган ҳамда кенг имкониятларга эга мукаммал framework ҳисобланади.

Willie Wheeler

Фаолиятимнинг дастлаб йилларида мен EJB 1.0 дан фойдаланиб бир нечта йирик корхоналарнинг турли лойиҳаларида иштироқ этдим. Java дастурлаш тилида корпоратив ишланмаларни яритшда оғриқли нуқталар мавжудлиги тез орада барчамизга маълум бўлди. Ўша пайтдаги мавжуд ечимлар кўп ҳолларда мурккаб, бироз зерикарли, ўрганиш узоқ вақт талаб қиладиган ва хатоликларга мойил эди. Ва ниҳоят 2002-йилда Rod Johnson тамонидан «Expert One-on-One J2EE Design and Development» номли китоб чоп этилди, ўша пайтда мен ўзгаришлар ва янгиликлар учун аллақачон тайёр эдим. Rod ўз китобида келтириб ўтган ва кейинчалик Spring Frameworkга киритилган мазкур ғоялар нафақат менда, балки бутун Javaда дасурчилари ҳамжамияти орасида ҳам катта таъсурот қолдирди. Мазкур framework бутун бир инфратузилма кодини кўриб чиққанлиги сабали, менинг кодларим ҳам тоза ҳам содда ва энг асосийси ҳатоликлар жуда кам учрар эди. Шундай қилиб Springдан фойдаланган ҳолда иловаларни ишлаб чиқиш менга самаралироқ ва завқлироқ бўлишини ҳис қила бошладим. Шу пайтдан бошлаб мен Spring Framework ҳақидаги барча янгиликларни қизикиш билан қарши оладиган бўлдим.

Spring ўз навбатида шиддат билан жадал суръатлар билан ривожлана бошлади ва кундан-кунга улғайгани сайин мазкур framework атрофидаги технологиялар ва экотизим ҳақидаги билимларни ўрганишга мендаги иштиёқ ҳам ортиб борди. Йиллар давомида камина мазкур технологиялар тўгрисда чуқур изландим ҳамда техник китобхон даражасига айландим. Шу тариқа мен бошқа муаллифларнинг китоблари билан ҳам танишиб чиқдим ҳамда тез орада ўз билимларимни техник маълумотлар билан бойитиб боришни бошладим. Орадан маълум бир вақт ўтиб Manning нашриёти менга мазкур китобнинг муаллифи ҳисобланган Willie билан ҳамкорлик қилиш имкониятини тақдим этди. Бу менинг учун катта имкониятларни яратиб берди, Willie билан ҳамкорликни йўлга қўйишимдан олдин гўёки мен худди улкан бир деворнинг нариги тамонида бўлганимни ҳис қилдим.

Joshua White

**Миннатдорчилик**

Мазкур китобни ёзишда менга якиндан чин кўнгилдан ёрдам кўрсатганларнинг барчасига чуқур минтдорчилик билдираман. Мен узоқ йиллик китобхон сифатида китоб муаллифлари ўз яқинларига, фарзандларига ва ўз атрофидаги инсонлар учун қилинган ноқулайликлар учун билдирилган миннатдорчилигини жуда кўп ўқиганман. Лекин айни дамда айнан ёзувчи ўриндиғида ўтирган ҳолатимда мазкур миннатдорчиликни ифодалаш учун қанчалик кучли иборалар орқали этироф этилган тақдирда ҳам, аслида буни тўлиқ ифодалаш қанчалик қийин эканлигини янада яхшироқ ҳис қилмоқдаман.

Энг биринчи бўлиб менга жуда катта ёрдам берган инсонга, яни менинг рафиқам Rayleneдан жуда катта миннатдорман, у мендан ўзи учун ҳеч нима талаб қилмади балик мазкур саъй-ҳаракатларимни тўлиқ қўллаб қувватлади. Бизнинг шахсий ҳаётимиз билан боғлиқлиги сабали, аёлим менинг номимдан қилган кўпгина қурбонликларини очиқлашимга ҳалақит қилади. Лекин шуни аниқ айтишим мумкинки, рафиқам мазкур китобнинг нусҳасини қўлларига олишни жуда ҳам интиқлик билан кутмоқда, ана ўшанда бу унга ёқимли бўлиши мумкин. Raylene сиз менга ушбу китобни ёзишим учун имкон яратиб берганингиз учун катта рахмат айтаман! Ушбу китобнинг муқовасида худди мен каби сенинг исминг алоҳида аҳамият касб этади.

Кейинги ўринларда мен ўз фарзандларимиз Max, Jude, Lucy, ва Benга катта рахмат айтаман, уларнинг иккитаси хеч қачон ҳаётларини «китобсиз» тасаввур қила олишмаган. Улар ҳам ҳудди оналари каби, бутун жараён давомида сабр ва тоқат билан мени қўллаб қувватлашди мендан ортиқча хеч нарсани сўрашмас эдилар ва мен улардан жуда ҳам миннатдорман.

Қайнота ва қайнонам Ray, Jane, Renee, Rana, ва Raymondларга кўплаб дам олиш кунларида меинг оила аъзоларимга дам олишлари учун ўз уйларини тақдим этишганликлари учун алоҳида рахмат айтаман.

Ва албатта менга ҳамкорлик қилган мазкур китобнинг ҳаммуаллифи Joshга ушбу китобни охириги етказиб кўйишда жуда катта ёрдам бергани учун ўз миннатдорчилигимни билдираман.

Шунингдек ўзим ҳамда Josh номидан Manning нашриёти жамосига ўз миннатдорчилигимни билдириб ўтаман, жумладан:

Marjan Bace, Cynthia Kane, Elizabeth Martin, Mary Piergies, Maureen Spencer, Tiffany Taylor, and Megan Yockey.

Ривожлантириш ишлари бўича муҳарриримиз Cynthia Kaneга эса алоҳида ўзимнинг самимий рахматимни айтмоқчиман.

Таҳририят тамонидан менга жуда катта ёрдам кўрсатилишидан ташқари яна бир инсон Cynthia ушбу лойиҳани мувоффақиятли якунлашда ҳаракатлантирувчи куч бўлди. Cynthia сизнинг бой тажрибангиз, сабр-тоқатингиз ва мени доимо қўллаб-қувватлаганингиз учун рахмат.

Қолаверса теҳник шарҳловчиларга ҳамда ишланмаларни ишлаб чиқиш жараёнида билдирган билдирилган фикр-мулоҳазалари ва йўллаган бебаҳо саволлари учун MEAP мижозларига рахмат. Уларнинг саъй-ҳаракатлари мазкур китобни янада яхшироқ ва мазмундор китобга айлантирди, жумладан:

Al Scherer, Brian O’Shea, Carol McDonald, Craig Walls, Daniel Alford, Deepak Vohra, Dmitry Sklyut, Erwin Vervaet, George Franciscus, Gordon Dickens, Jeremy Flowers, Jeroen Nouws, John Guthrie, John Ryan, John Tyler, Kenneth Kousen, Kenrick Chien, Mario Arias, Patrick Steger, Prasad A Chodavarapu, Rama Kanneganti, Ricardo Lima, Rizwan Lodhi, Robby O’Connor, Robert Casazza, Robert Hanson, Ryan Stephens, Srikanth Balusani, and Willem Jiang.

Шунингдек теҳник муҳарриримиз Doug Warrenга алоҳида рахмат айтаман, унинг тинимсиз саъй-ҳаракатлари ва тафсилотларга жуда этиборли бўлганлиги сабали китоб сезиларли даражада яхшиланди ва маъзмунан бойитилди.

Азизларим сизнинг ёрдамингиз бўлмаганида биз буни уддалай олмас эдик!

**Spring ҳақида**

Дастлабки даврларда Spring технолгияси иловаларни ишлаб чиқиш Java Enterprise Edition (JEE)ни соддалаштириш усули сифатида ишлаб чиқилган, лекин аслини олиб қарасак Spring бу қадар оддий framework эмаслигини унитмаслигимиз лозим бўлади. Аслида Spring frameworkнинг асоси ўнлаб сондаги улкан портфолияга эга бўлган турли ҳил лойиҳалар мазкур ядрони кенгайтириши билан қаторда бир нечта муҳим элементларни ҳам ўз ичига қамраб олган, жумаладан:

Хавфсилик, веб оқимларни жўнатиш, SOAP веб хизматлари(REST веб хизматларнинг асосий қисми ҳисобланади, корпоротив интеграциялар, маълумотларга пакетли ишлов бериш учуллари, мобил қурилмалар, турли ижимоий тармоқлар(Facebook, LinkedIn, Twitter, GitHub ва бошқлар), турли ҳил NoSQL маълумотлар базаси билан ишлаш(MongoDB, Neo4j, Riak ва бошқалар), шунингдек BlazeDS/Flex, AMQP/Rabbit ва бошқалар. Агарда «оддий» дейилганда бир нечта қисмдан иборат эканлигини англатадиган бўлса ҳамки Spring шунчаки оддий эмас.

Шу ўринда бир ҳақиқатни яхшилаб англаб етмоғимиз зарур бўлади, яъни Spring тузулиши жиҳатидан мураккаб, аммо унинг ёрдамида иловаларни ишлаб чиқариш жуда ҳам соддароқ амалга оширилади. Махсус воситалар ёрдамида бажарилиши лозим бўлган ишлар енгиллик билан амалга оиширлади. Бунга мисол сифатида IDEларни келтиришимиз мумкин бўлади.

Биз учун энг қувонарли тамони шундаки, Spring ўрганишимиз осон бўлиши учун турли ҳил усуллар ёрдамида янгиликларни ўз вақтида ёритиб боради. Бунга мисол қилиб қуйидагиларни келтиришимиз мумкин:

* Мазкур framework ядроси ва унинг портфеллари орасида бир нечта фарқлар мавжуд. Лойиҳанинг бошланғич тузилмаси яратилаётган ишланмалар учун зарур бўладиган умумий талабларини қаноатлантиради, бунга мисол сифатида маълумотлар базасини ишлаб чиқиш, объектлар ва улар орасидаги боғланишларни тасвирлаш, тразакциялар, веб ишланмалар ва бошқаларни айтишимиз мумкин. Шуни эсда сақлашимиз лозим бўладики портфел лойиҳалари асосан махсус мақсадлар учун мўлжалланган бўлади. Шунинг учун ҳам Springни ўрганишнинг энг самарали усулларидан бири эса дастлаб frameworkнинг ядросини ўрганиб чиқиш, ундан кейин эса портфел лойиҳаларини ўрганиш ҳисобланади.
* Бир нечта ёндашувлар ва андозалар билан ишлаш, бунга мисол қилиб POJO, мустақил инекциялар, шаблонлар, AOP аосланган автоматик прокциялаш ва бошқалар бутун бир framework тузилмаси бўйлаб бир неча бор такрорланади. Springни ўрганиш аслида оқиллик билан чегаралаб олинган асосий ёндошувлар тўпламини ўрганиш масаласидир.
* Энг дастлабки ёндашувлардан бири XML ёрдамида ишлаб чиқилган кофигурация ҳисобланар эди. Орадан бироз муддат ўтиб, бази сабаларга кўра дастурчилар ўз шикоятларини билдиришни бошлашди. Келинг ушбу шикоятлар билан танишиб чиқамиз: лойиҳа учун ёзилиши лозим бўлган конфигурация жуда ҳам аниқлик ва узоқ вақт талаб қилар эди, шунингдек XML форматини унинг муқобили бўлган JSON, YAML форматлари билан таққосланганда ҳам мувофақиятсизликка дуч келар эди, эҳтимол бу ишлар учун Javaнинг мавжудлиги ҳам бундай шикоятлар юзага келиши сабаб бўлган бўлиши мумкин. Spring мазкур муаммони бир қанча содда конфигурация вариянтини қўшиш орқали ҳал қилди. Жумладан namespaceларни қамрови орқали, annotationлар асосида ҳамда Java ёрдамида.
* Spring дастурчилари жамоси Java оламидан ташқарида бўлаётган воқеа ва янгиликларга дойимий эътибор қаратишади, шунингдек иловаларни ишлаб чиқаришни соддалаштириш учун билдирилган таклифлар ва ғояларни маъмнуният билан қабул қилади. Айниқса Rails иштирок қилган Railsда бундай ғоялар билан бойитилган манбалардан бири ҳисобланади. Spring Roo ва Grails аслида Railsнинг клони ҳисобланади, шунинг ўз навбатида худди Rails ўхаш бўлган ишланмаларни Java and Groovyга ёрдамида амалга оширишни таъминлайди. SpringSource кодлари ва манбалари ишлаб чиқишда Groovy дастурлаш тили ҳамда Grails номли frameworkлари асосий ўринларда туради.
* Spring шунингдек Spring Tool Suite (STS) номи остидаги IDEни жуда ҳам яхши қўллаб-қувватлайди, Eclips бренди остида чиқарилган ҳам кенгайтирилган версиларида ишлайди. Бунинг ёрдамида комонентларинг ўзаро боғланишларини тасвирлашни енгиллаштиради, бу эса бизга аспектга йўналтирилган дастурлаш бўйича маслаҳатлар ва улардан қандай қўлланилишини тушунимизни ва шу кабиларни анча енгиллаштиради. Баъзи бир портфел лойиҳалари IDEлар учун қўимча интеграцияларга эга, масалан Spring Integration ва интеграциялар визуализацияси бунга ёққол мисол бўла олади.

Бу санаб ўтганларимиз тўлиқ рўйхат эмас, лекин бунинг ёрдамида ўзимиз учун зарур бўлган фикрларни англаб олишимиз учун етарли ҳисобланади.

Мазкур китобни ёзишда биз ўрганувчилар учун имкон қадар содда ва тушунарли қилиб етказишга ҳаракат қиламиз. Бунинг учун эса бирон бир вазиятда юз бериши мумкин бўлган кенг кўламдаги муаммоларни тақдим этамиз(агарда сиз бундай муаммоларга дуч келмаган бўлсангиз) ва албатта уларни ҳал қилиш учун Spring бизга қандай ёрдам бериши мумкинлигини ҳам кўрсатиб ўтамиз. Мазкур frameworkнинг ядроси ўрганиш мазкур китобни барча жойларида фойдаланилган, шунинг учун сизда янада кўпроқ амалиётга эга бўлишингиз мумкин. Шу билан бир қаторда биз сизлар билан бир нечта портфел лойиҳаларини ишлатиб кўрамиз, сабаби улар мавжуд муаммолар ҳал қилиш учун қўпроқ мос келади, қолаверса уларни такрор-такрор ишлатиш орқали ўзингиз керакли мавзуларни тушунишингиз учун ёрдам беради.

Азиз китобхонлар сиз ушбу китобни ўқиб тугатганингиздан кейин сиз frameworkнинг ядроси ва бир нечта портфел лойиҳалари ҳақида кенг тушунчаларга эга бўласиз. Сиз ортирган тажрибаларингиз асосида сиз ҳам Spring ёрдамида реал лойиҳаларни амалга ошириш учун етарли даражада билим ва кўникмаларга эга бўласиз. Шуни эсингиздан чиқармаслик лозимки биз сизни бу борада эксперт даражасига чиқара олмаймиз, лекин энг зарури ҳисобланган асосий саволларга жавоб топишни ва уларни каердан излашингиз мумкинлигини етарли даражада тушунишингиз ҳамда уларнинг маъзмун ва моҳиятини англаб етасиз.

**Китоб ҳақида**

Мазкур «Spring in Practice» аслида амалиётга йўналтирилган китоб ҳисобланади. Китобнинг дастлабки 3 та боби асосан асосий назарий маълумотлар ҳақида келтирилган, унда энг муҳим маълумотлар ёритилган кўпчилик ўрганувчилар ўзлари учун янги билимларни эгаллашларини умид қиламиз. Китобнинг 2-боби ҳамда охирги боб ҳисобланган 11-бобларда берилган мўаммоларни ҳал қилишни тартиб билан, босқичма-босқич техник ечимлари билар берилган, унда cookbook форматидан фойдаланилган. Ушбу китобда 66 турдаги техника усуллар ёритилган.

***Йўл ҳаритаси***

Юқорида таъкидлаб ўтилганидек, 1-бобдан бошлаб 3-бобгача асосан кейинги боблар учун ҳам фойдаланиладиган энг асосий материаллар тақдим этилган. Улар куйида келтирилган:

* 1-бобда асосан енгил бошкаралидагина контейнернинг Spring ёрдамидаги инверсияси ёритиб берилган.
* 2-бобда маълумотларга рухсат олиш, ORM ва транзакциялар билан қандай ишалаш ҳақида батафисл маълумотлар келтирилган. Шунингдек биз бу ерда асосан умумий қабу қилинган(generic) DAOни амалга оширишнинг ананаий ёндашувини қўллаган бўлсак ҳам, шу билан бир қаторда биз Spring Data ва JPAнинг кучли ва биз учун самарали бўлган имкониятларидан фойдаланиш ҳақида ёритиб ўтилган.
* 3-бобда эса ваб иловаларни ишлаб чиқиш улкан имкониятларга эга бўлган Spring Web MVCдан қаданй фойдаланиш тўғрисида маълумотлар акс этагн. Шунингдек мазкур бобда мобил иловаларни ишлаб чиқишни қўллаб қувватлаш мақсадида Spring Web MVC имкониятларини кенгайтиришга ёрдам берадиган Spring Mobile ҳақида ҳам сўз юритилган. Бур ерда Spring Mobile энг асосий ўринни эгалламаган бўлишига қарамасдан бу бобга жуда ҳам мос келади, шу боисдан ҳам биз ундан фойдаланишни ҳам кўрсатиб ўтдик.

Кейинги иккита бобларда, яъни 4-боб ва 5-бобларда эса рўйхатга олиш формаси ишлаб чиқиш учун икки хил ёндашуви кетлитилган.Мазкур тақдим этилган маълумотлардан бошқа турдаги муаммоли соҳалар учун қўллаш жуда ҳам осондир:

* 4-бобда веб муҳитларда ишлашга учун мўлжалланган ягона-саҳифа тамойилига асосланган ҳамда Spring Web MVC дан фойдаланган ҳолда рўйхатга олиш формасини қандай ташкил қилиш мумкинлиги батафсил кўрсатилган. Мазкур усул одатда ягона сахифадан иборат бўлган формаларни ишлаб чиқишда қўлланилади.
* 5-бобда янада мураккаброқ, кўп саҳифалардан ташкил топган рўйхатга олиш жараёнини амалга оишириш учун Spring Web Flowдан фойдаланилган. Бу усулдан эса бир нечта босчиқлардан иборат бўлган ва умумий мақсадга қаратилган формаларни ишлаб чиқишда фойдаланилади. Мисол сифатида даттурдаги турли жараёнлар, тўлов жараёнларини келтиришимиз мумкин.

Кейинги бобларимизда, яъни айнан 6-боб ва 7-бобда маълумотларнинг хафсизлигини таъминлашнинг иккита муҳим жиҳатини батафсил куриб чиқамиз, улар орасида фарқларни таққослаб кўрамиз:

* 6-бобда тизимга кириш жараёнига Spring Security жорий қилиш ва ундан қандай қилиб фойдаланиш ёритилган.
* 7-боб ҳам Spring Securityнинг мантиқий давоми ҳисобланади ва веб-форум иловасига фойдаланувчилар учун авторизацияни кандай қўшиш мумкинлиги ҳақидаги фойдали маълумотлар келитрилган. Шунингдек фойдаланувчилар роли асосида авторизациядан ўтиш жараёнларини кўриб чиқамиз.

8-боб ҳамда 9-бобларда иловаларни ишлаб чиқишда жуда ҳам кенг тарқалган муаммони ўрганиш ва уни таҳлил қилишга қаратилган, яни бу ерда фойдаланувчиларнинг бир бирлари билан мулоқот қилишларини амалга ошириш усуллари ҳақида бўлади.

* 8-бобда фойдаланувчилар учун мурожат формалари, электрон почта хабарлари, шунингдек билдиришномалар, электрон почта манзиллари рўйхати, RSS хабарлари учун махсус ленталар қандай амалга ошириш мумкинлигини ўз ичига қамраб олган.
* 9-бобда эса кўпчилик учун машхур бўлган StackOverflow тизимида ишлатиладиган Page-Down матн мухаррири ва унинг имкониятларидан қандай фойдаланиш уни амалга оишириш ҳақида бўлади.

Қолган бобларда эса асосан алоҳида биз учун фойдали бўладиган мазвулар ёритилган:

* 10-бобда яратилган дастурлар учун тестлар ишлаб чиқиш, хусусан интеграция тестларини ишлаб чиқиш Spring TestContext Framework дан қандай фойдаланиш кўрсатилган.
* 11-бобда маълумотлар базасини бошқариш учун зарур бўладиган махсус конфигурациялар кўрсатиб ўтилган. Улар қаторига Neo4j, Spring Data Neo4j, Spring Social ва бошқалар киради.
* 12-бобда мақолаларни тақдим этиш механизмини қандай ташкил этиш ҳамда уларда Java Content Repository (JCR) ва MongoDBдан фойдаланиш усуллари кенг ёритилган.
* 13-боб Spring ёрдамида мурожатларни қайта ишлаш тизими ишлаб чиқишни ўз ичига олган. Бу ерда асосан илованинг тузилмасини ишлаб чиқишга кўпроқ урғу қаратилган.
* 14-бобда Spring ёрдамида сиз ўзингизнинг махсус frameworkингизни ишлаб яратиш учун зарур бўладиган усуллар қўрсатилган, шунингдек namespaceга асосланган конфигурациялар ва AOP ва аннотацияларни ва бошқаларни қўллаб қўвватлаш ҳақида маълумотлар ўрин олган. Бизнинг мисолимизда мазкур framework сайтинг бардошлилигини таъминлашга қаратилган.

Иловада эса мазкур киитобда ишлатилган дастур кодларини қандай ташкил қилганимиз, шунингдек уни қандай яратиш, созлаш ва уни ишга тушириш мумкинлиги ҳақида тушунтириб ўтилган. Сиз ушбу маълумотлар билан танишиб чиқиб китобда келтирилган дастур кодларини ишга тушириб, ўз билим ва малакаларингизни оширишингиз мумкин бўлади

***Ушбу китобдан кимлар фойдаланиши мумкин?***

Мазкур китобнинг номидан ҳам кўриниб турибдики, «Spring in Practice» Sprin Frameworkидан амалда қандай фойдаланишни ўрганиш учун ёрдам беради. Гарчи биз вақти-вақти билан зарурий бўлган билан тушунчалар ва ёндашувларни тушунтириб ўтган бўлсак ҳам, у аслида камлик қилади. Шунинг учун сиз ўз билимларингизни янада бойитиш устида меҳнат қилишингиз лозим бўлади. Бу ерда биз асосан бирон бир нарсани қандай қилиш мумкин эканлигини ва мавжуд усулларини кўрсатиб ўтишга ҳаракат қилганмиз.

Шунингдек ушбу китобда Spring нима ва нима учун керак эканлиги борасида етарли даражада тажрибага ортиришингизга умид қиламиз. Бу Spring ҳақидаги биринчи китоб эмас балки, Spring асосларини янада чуқурроқ ўрганишга қартилган китоблар учун фойдали ва манфатли бўлган қўшимча бўлади. Бунга мисол сифатида 2011-1илда Manning нашриёти тамонидан Craig Walls муаллифлиги асосида чоп этилган «Spring in Action» китобининг учинчи нашрини келтиришимиз мумкин бўлади.

Мазукр китобдаги деярли барча мисоллар кенг маънода веб иловаларни ишлаб чиқишга қаратилгандир. Бу ерда веб иловаларни ишлаб чиқишнинг энг муҳим жиҳим жиҳатлари, шунингдек муаллифнинг бой тажрибалари ўрин олган. Шу билан бир қаторда сиз аллақачон Javaда веб иловаларни ишлаб чиқиш асосларини ҳисобланган HTTP, servlets, JSP, тэглар кутубхонаси билан танишиб эканлигингизни умид қиламиз.

Кўпгина мисолларда корпоратив APIларни ишлаб чиқишда JNDI, JPA, JavaMail, ва JMX лардан фойдаланилган. Шунингдек мазкур мисолларда Hibernateнинг имкониятларидан ҳам кенг фойдаланилган. Бундан ташқари ҳозирги кундаги энг сўнги тенденциялардан ҳисобланган мобил қўрилмалар, ижтимоий тармоқлар ва NoSQLларни корпортив ва нокорпоратив созламалари келитирилган, шунингдек баъзи мисолларда GitHub, OAuth, MongoDB, ва Neo4j каби мавзуларга ҳам алоҳида этибор қаратилган. Умуман олганда юқорида санаб ўтилганларни сиз ўзингиз мустақил ўрганишингиз учун етарли даражада тажрибага эгасиз деб ҳисоблаймиз(ҳатткоки сиз уларни бошқа китоблардан ҳам ўрганишингиз мумкин) шу боисдан биз мазкур китобда асосан Spring билан ишлаш тамонларига кўпроқ аҳамият қаратмиз.

Шу ўринда битта муҳим нарсага этибор қаратишимиз лозим бўлади, яни мазкур китоб дастурлашни эндигина ўрганишни бошлаган бошловчилар учун эмас, балки сўнги бир неча йиллар давомида Javaда дастурлар ишлаб чиқиш билан шуғулланадиган кўпчилик сондаги дастурчилар мазукр киитобдан Spring ҳақидаги ўз билимларини кенгайтиришлари учун фойдали деб топишлари мақсадга мувофиқ ҳисобланади.

**Дастур кодлари ва уларни юклаб олиш ҳақидаги келишувлар**

Мазукр китобда келтирилган барча миссоллар учун дастур кодларини [www.manning.com/SpringinPractice](http://www.manning.com/SpringinPractice) ёки <https://github.com/springinpractice> сайтида топишингиз мумкин. Мазукр китобда келтирилган кодлар китобдаги бобларга мос равишда sip02, sip03, sip04 ва бошқалар каби репозиториялар шаклида тақдим этилган. Иловада эса дастурдаги кодларни яратиш, созлаш ва уни қандай қилиб ишга тушириш ҳақида батафсил маълумотлар келтирилган.

Кўийда келтирилган шартлар ва келишувлардан фойдаланилган:

* Italic шрифти янги атамаларни белгилаш учун ишлатилади
* Courier шрифти дастур наъмуналарини белгилаш учун, шунингдек элементлар, хосслар, метод номлари, синфлар, интерфейслари каби бошқа идентификаторларни белиглашда ҳам ишлатилади
* Дастур коди учун изоҳлар кодларни бир нечта кисмларида ишлатилади, баъзи бир изоҳлар рақамланган ва ўқларга эга чизиқлар билан ҳам белгиланади. Мазукр изоҳларга код тугалланганганидан кейин алоҳида тушунтиришлар бериб ўтилган.
* Кодлар каторининг давомийлигини белгилаш учун махсус белги ➥ ишлатилади

**Муаллифнинг интернет тармоғида форуми ҳақида**

Азиз китобхон сиз мазкур «Spring in Practice» китобини харид қилишингиз биланоқ Manning нашриёти тамонидан бошқариладиган веб-форумга бепул кириш имкониятини ҳам кўлга киритасиз. У ерда сиз мазкур китоб ҳақида ўз фикр ва мулоҳаларингизни билдиришингиз ҳам узингизни қизиқтирган саволларингизга муаллиф ва бошқа фойдаланувчилар тамонидан жавоб олишингиз мумкин. Форумга кириш ва унга обуна бўлишингиз учун веб броувзерингиз манзилиги [www.manning.com/SpringinPractice](http://www.manning.com/SpringinPractice) киртишингиз талаб этилади. Бу рада рўйхатдан ўтганингиздан кейин сизга мазкур форумга қандай кириш мумкинлиги, қандаё ёрдам олиш мумкинлиги ҳамда форумда иштирок этувчилар учун одоб-ахлоқ қоидалари тақдим этилади.

Шу ўринда Manning нашриёти уқувчилар ва муаллифлар олдидиаги маъсулиятини тўлиқ ҳис этади ҳамда ўқувчмлар ва муаллиф ўртасидаги мазмундор мулоқотни таминлаш мумкин бўлган қулай мўхитни яратиб беришдан иборатдир. Шунингдек муаллифларни иштроклари учун ҳеч қандай ҳақ тўланмайди, балки мазкур форумга кўшган мажбурий ҳиссаси ҳисобланади.

Муаллифнинг қизиқизларини ортириш мақсадида уларга бир нечта қизиқарли ва мўраккаб саволларни беришингизни тавися этамиз. Ҳозирда «Author Online» форумида ҳам олдини муҳокама қилинган савол-жавоблари архивидаги маълумотлар асосида китоб чоп қилиш ишлари бошлаб юборилган, унгача сиз нашрийт веб сайтидан фойдаланиб туришингиз мумкин бўлади.

**Муаллифлар ҳақида**

WILLIE WHEELER ҳозирги кунда Expedia комапаниясида иловалар ишлаб чиқиш бўйича бош дастурчи муҳандис лавозимида ишлаб келмоқда, шун кунларда у веб ёрдамида иловаларда кўшимча имкониятларни ишлаб чиқиш ҳамда уларнинг узлуксиз ишлашини таъминлашга кенг этибор қаратиб келмоқда. Шунингдек у 1997 йилдан бошлаб Java дастурлаш тилини билан ишлашни бошлаган бўлса кейинчалик яъни, 2005-йилдан бошлаб Spring билан ишлаб келмоқда. Агарда сизга мазкур китоб мақул бўлган бўлса, худди мана шундай қизизқарли маълумотларни springinpractice.com блогидан ҳам излаб топишингиз мумкин. У шунингдек zkybase.org/blog да ўзининг devops ҳақидаги блогини ҳам юритади. Айни дамда Willie рафиқаси Raylene ва уларниг 4 нафар фарзандлари Max, Jude, Lucy, ва Ben билан Washington штатидаги Sammamish шахрида истиқомат қилади.

JOSHUA WHITE ҳозирда Cigna компаниясида дастурий инжинеринг ва инновациялар бўйича директори лавозимида ишлаб келмоқда. Шунингдек бир нарсани алоҳида таъкидлаш лозимки у ўз фаолияти давомида Молиявий ташкилотлар ва соғлиқни сақлаш ташкилотлари учун мураккаб турдаги дастурий махсулотлар ва уларнинг архитектурасини ишлаб чиқиш соҳаларида 12 йилдан кўпроқ тажрибага эга ҳисобланади. У Spring Framework яратилиши биланоқ 2002-йилдан бошлаб уни маъмнуият билан ундан фойдаланишни хуш курган. Айни дамда Joshua Connecticut штатининг Farmington шахрида истиқомад қилади.

**Китоб муқовасидаги тасвир ҳақида**

Мазкур «Spring in Practice» китобнинг муқовасида жой олган тасвирда Султони мухофаза қилувчи юқори мансабдаги «Буюк Signior Саркардаси» ифодаланган. Ушбу картина 1802-йилда William Miller тамонидан Лондоннинг Old Bond кўчасида **Усмонийлар империяси либослари тўплами** коллекциясидан олинган. Мазкур коллекциянинг тўлиқ сарлавҳаси қандай номлагани ҳақида маълумотлар мавжуд эмас, шу боис ҳам унинг асл сарлавҳасини топиш имкониятига эга эмасмиз. Китобнинг мундарижасида кўрсатилган тасвирларда ингилиз ва франсуз тилларида мазкур картина устида ишлаган иккита рассомнинг номлари кўрсатиб ўтилган. Албатта бу иккита рассомлар сизга тақдим этилаётган мазкур дастурлаш соҳасидаги китобнинг муқовасида ўз санъат асарларини кўриб шубҳасиз ҳайратда қолган бўлар эдилар. Албатта улар бундан қарийб икки сар олдин яшаб ижод қилишганини эслатиб ўтишимиз лозим.

Ушбу ноёб тўпламни Manning мухаррири Манхеттин шахрининг 26-чи Ғарбий кўчасидаги «гараж»даги аниквар махсулотлар сотиладиган бозордан харид қилган. Сотувчи Анқарада(Туркия) истиқомат қилувчи асли келиб чиқиши Америкалик бўлган сотувчи бўлиб, савдолашиш жараёни айнан куннинг охирида сотувчи ўз стундларини йиғиштраётган бир пайтда содир бўлган. Шу вақтда Manning мухарририда мазкур асарни тоиб олиш учун етарли миқдордаги нақд пул бўлмаган, у тамонидан кредит картаси ҳамда чек ёрдамида сотиб олиш таклифи ҳам сотувчи тамонидан хушмуомалалик билан рад этилган. Энг қизиғи сотувчи шу кеча қайтиб кетгани эса, мухаррирнинг мазкур асарни сотиб олишга бўлган умидларини янада сўндирди. Фақатгина битта умид қолган эди у ҳам бўлса эски услубдаги ўзаро қўл сиқиш билан савдолашилган ва барчаси оғзаки келишувдан иборат эди. Сотувчи пулни шунчаки унинг хисоб рақамига ўтказиб беришни таклиф қилди ва мухаррир қоғоздаги банк маълумотлари ўз портфелини қўлтиқлаганча ўз йўлида давом этди. Ҳеч қандай ортиқча гап сўзларсиз эртаси куни биз унинг ҳисоб рақамига зарур бўлган пул маблағларини ўтқаздик, шу ерда икккита нотаниш иккита шахснинг бир-бирига бўлган ишончи бизларда катта таъсурот қолдириши билан бирга бизга хурсандичилик тухфа этди. Бу бизга бир неча йиллар содир этилган ажойиб воқеани эслатиб туради.

Усмонийлар даврига оид либослар тўплами бизнинг бошқа китобларимиздаги тасвирлар каби бундан икки аср олдинги либослар, одатлар ва қадриятларнинг бойлиги ва уларнинг ранг баранглиги акс этдиради. Улар ўша даврнинг ва ҳозирги бизнинг давримизгача бўлган тарихий даврни эслатиб тўради. Худди шу даврдан бошлаб жамият орасида кийиниш қоидалари ўзгарди турли хил минтақалар ва табақалар орасида фарқлар йўққа чиқди. Бунинг натижасида турли минтақалардаги одамлар орасидаги фарқларни ажратиш қийинлашди ўзаро тенглик хукм сура бошлади. Қолаверса янада рангбаранг қизиқарли бўлган инсон интеллектларига асосланган техника ривожланган ҳаёт тарзига ўтилди.

Шу ўринда бизнинг Manning жамоси билан биргаликда ихтирочилик, ташббускорлик ва албатта компьютер соҳасидаги фаолиятнинг қай даражада қизиқарли эканлигини бундан икки аср олдингни минтақалардаги жамиятнинг ранг-баранг ҳаёт тарзини акс этдириш орқали намойиш этишни ҳамда уларни мазкур тўпламдаги тасвирлар орқали тасвирлаб беришга харакат килдик.

**1-боб. Spring билан танишув дарси.**

**Осон бошқариладиган контейнерлар инверсияси ҳақида тушунча**

Ушбу боб қуйидагиларни ўз ичига қамраб олади

* Spring Frameworkнинг асосий функционал соҳалари
* Боғланувчи инекциялар ёрдамида мослашувчан конфигурацияларни ишлаб чиқиш
* XML ёрдамида beanларни бир-бирига боғлаш
* Аннотациялар ёрдамида компонентларни сканерлаш ва атоматик боғланишни амалга оишириш

Мазкур бобда биз Spring Framework нима эканлигини ва уни муҳокама қилишдан бошлаймиз, унинг асосий қисмлари қисқача баён қилиб ўтамиз. Кейинги ўринларда Spring Frameworkнинг асосий тамойилларини кўриб чиқамиз ва бошқарув инверцияси ва унинг боғловчи инекциялар билан қандай боғланиши ҳақида гаплашамиз. Бобнинг якунида эса Spring Core Container дан ўз қўлимиз билан қадай фойдаланиш мумкинлиги кўрсатиб берадиган кичик мисолни кўриб чиқамиз. Сиз ҳам тайёр бўлсангиз у ҳолда биргаликда ўрганишни бошлаймиз.

* 1. **Spring нима ва ундан нима учун фойдаланилади**

Келинг дастлаб Spring нима эканлигини ҳақида сўз юритамиз, қисқача қилиб айтганда Spring Framework - бу Java дастурлаш тилида корпоратив дастурий таъминотларни ишлаб чиқишни соддалаштириш учун ишлаб чиқилган очиқ кодли дастур ҳисобланади. Мазкур ишларни амалга оширишда дастурчиларга ушбу framework ўзининг компонент модели ва соддалаштирилган ва ихчам API интерфейслар тўпламини тақдим этиш орқали амалга оширилади. Бунинг натижасида дастурчилар мураккаб турдаги иловаларни ишлаб чиқиш жараёнида юз бериши мумкин бўлган қийинчиликлар ва дастур кодларидаги юзага келиши мумкин бўлган хатоликларнинг олдини олади, яни улардан дастурчиларни самарали изолация қилади.

Сўнги тўққиз йил давомида мазкур frameworkнинг имкониятлари ва қамрови сезиларли даражада кенгайиши билан бир қаторда уни ўрганиш ва ишлатиш осонлашди. Ушбу framework 20 дан ортиқ модулларга эга бўлди, биз уларни асосий ҳисобланган 6 та функционал соаҳларга ажратишимиз мумкин. Қуйида тақдим этилган 1.1-расмдаги тасвирланганидек, бу функционал соҳалар маълумотлар учун руҳсат олиш/интеграцияси, Веб, Аспектга йўналтирилган дастурлаш(AOP), Ускуналар, Контейнер ядроси ва Тестлаш каби соҳалардан ташкил топган. Ушбу модулларга ажратиш орқали ўз иловаларини ишлаб чиқмоқчи бўлган дастурчиларга бутун бир frameworkни кўшмасдан балки ўзлари учун керакли бўлган қисмларни эркин танлаш имкониятини тақдим этади.

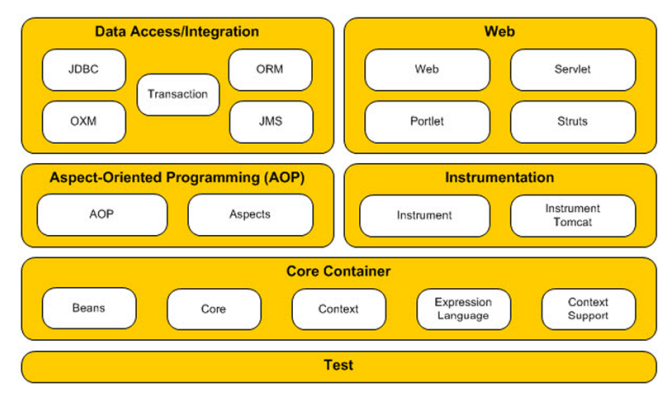
Келинг кейинги саёҳатимизни ушбу функционал соҳаларнинг ҳар бири билан танишиб чиқишдан бошлаймиз.

* + 1. **Frameworkнинг асосий таркибий қисмлари**

Ушбу параграфда биз сизларга Spring нинг олтита функционала соҳаларинининг ҳар бири ҳақида қисқача маълумот бериб ўтамиз. Шунингдек китобнинг кейинги қисмларида берилган махсус мисоллар ёрдамида биз ушбу мавзуларнинг ҳар бирига янада чуқурроқ кириб борамиз.

**Spring контейнернинг ядроси ҳақида**

Биз кейинги 1.2-бўлимда боғланувчан инекция (DI - dependency injection) нима эканлиги ҳақида батафсил кўриб чиқишга ҳаракат киламиз, Ҳозирча эса боғланувчан инекция бу Spring Frameworkнинг асосини ташкил этишини ҳамда бошқа барча модулларнинг фундаментал имкониятларини тақдим этишини билишимиз ва унинг англаб етишимиз кифоя қилади. Мазкур контейнер сизнинг иловангиздаги мавжуд дастурнинг коди билан, уни яратиш, созлаш ҳамда bean компонентлар(бу ҳақида кенроқ алоҳида тўхталиб ўтамиз)ни бошқаришни алоҳида ажратиш учун зарур бўлган имкониятлар тақдим этади.



***1.1-расм. Мазкур тасвирларда Springнинг олтита функционал соҳасини акс этдирувси умумлаштирилган диаграмма келитрилган***

**Аспектга йўналтирилган дастурлаш(AOP) ҳақида**

Мазкур Spring Framework аспектга йўналтирилган дастурлаш икки хилда, яъни Spring AOP деб номланувчи анчагина содда ёндашув асосида, шунингдек мураккаб ёндашувга асосланган AspectJ ёрдамида қўллаб қувватлайди. Модулларга ажратиш ва ундан такроран фойдаланиш хусусиятларини сақлаб қолиш билан боғлиқ муаммола(хавфсизлик, логлар рўйхати, транзакцияларни бошқари ва бошқалар)ни бартараф этишни ўз ичига қамраб олишга қаратилгандир. AOP билан кейинчалик бу ҳақида батафсил тўхталиб ўтамиз. Кўпчилик ҳолатларда бу муаммоларни тизимнинг қолган муаммоларидан алоҳида ажратиш имкони бўлмайди, бунинг натижаси дастур кодларида такрорланишлар ва тизимлар орасида сзеиларли даражадаи боғлиқликлар вужуда келиши мумкин. Боғланувчи контейнерлари каби, AOP ҳам frameworkнинг турли қисмларидан мустақил равишда фойдаланишини қўллаб қувватлаши оркали дастурчилар учун фойдали ҳисобланади. Мисол сифатида Spring AOP декларатив транзакцияларни бошқаришни тўлиқ қўллаб-қувватлайди, чунки транзакциялар бир-бири билан боғлиқ ҳолда ишлайди.

**Маълумотлар учун руҳсат олиш/интеграциси**

Мазкур маълумотларга кириш ва интеграция модули Java ёрдамида маълумотлар базасига уланиш учун мўлжалланган API(JDBC), Объект боғланишлари харитаси(ORM), Объект/XML боғланишлари харитаси(OXM), Java хабарлар хизмати(JMS) ҳамда транзакцияларни қўллаб-қувватлайди.

JDBC модули маълумотлар базасига уланиш ва уланиш учун махус йўлаклар(pools)ни автоматик бошқариш орқали, шунингдек дастурчилар хатога ўл куйиши мумкин бўлган хатоликлардан холи бўлган код ёзиш имкониятини берадиган абстракция қатламини тақдим этади. Бундан ташқари java.sql.ResultSets объектларни рўйхатга жойлаштиришни ҳамда маълумотлар базасидаги ички процедураларни ишга тушриш имкониятини осонлаштиради.

Агарда сиз маълумотлар базаси билан ишлаш учун тўғридан-тўғри JDBC дан фойдаланмасдан, унинг ўрнига ORMдан фойдаланишни афзал кўрсангиз, демак бунинг ҳам имкони мавжуддир. Ўз ўрнида ORM модули Hibernate, iBATIS, Java Data Objects (JDO) ва Java Persistence API (JPA) каби машхур ORMларни ҳам қўллаб-қувватлайди.

iBATIS ҳақида қисқача эслатма. Apache iBATIS дан фойдаланиш 2010-йилда расмий равишда тугатилган, унинг ўрнини MyBatis(mybatis.org) эгаллаган. Гарчи iBATIS 2 версияси Spring 2 ишлаб чиқарилган вақтда қўллаб-қувватланган бўлсада, Spring 3 ишлаб чиқарилишдаги вақтлар оралиғи ҳисобига ҳалигача расмий қўллаб-қувватлашни ҳозирчалик ўз зиммасига олмайди. MyBatis-Spring ҳақида қўшимча маълумотларга эга бўлиш учун қуйидаги [www.mybatis.org/spring](http://www.mybatis.org/spring) манзилга ўтинг.

OXM модули эса Java Architecture for XML Binding (JAXB), JiBX, XMLBeans, and Xstream учун Castor сифатида Объект/XML боғланишлари харитаси билан ишловчи воситаларни соддалаштириш ҳам яна бир нечта фойдали таклифлар билан биргаликда абстракция қатламини тақдим этади.

JMS модули хабарларни яратиш ва улардан фойдаланиш учун соддалаштирилган APIларни тақдим этади. Нихоят транзакция модули ҳам прагматик транзакцияларни ҳам декларатив транзакцияларни бошқаришни қўллаб-қувватлайди.

**WEB ҳақида**

Springнинг Web модули аосан иловаларга Springни интеграция қилиш, қўп қисмли файлларни юклаш ҳамда уларни масофадан бошқариш имкониятини тақдим этувчи умумий веб инфратсруктура кодини тақдим этади. Шунингдек у бизларга сервлетлар ёки портлетларга асосланган Model-View-Controller (MVC) frameworkидан ташқари яна бир нечта Struts, JavaServer Faces (JSF), Velocity, FreeMarker, and JavaServer Pages (JSP) каби машхур вебга ўналтирилган frameworkлар ва технологиялар билан биргаликда ишлаш имкониятини беради.

**Тестлаш ҳақида**

Мазкур модул энг охирги ўринда жойлаган бўлса ҳам мазкур модул framework энг муҳим ҳисобланган Spring иловаларини тестлаш ва синовдан ўтказиш имкониятини тақдим этади. Ушбу модул JUnit ва TestNG frameworkларидан фойдаланишни қўллаб-қувватлайди.

Шу билан биз сизларга умумий олиб қаралганда Spring Framework кандай кўринишини тақдим этдик, келинг энди мазкур frameworkдан фойдаланишнинг ўзига хос афзалликларини муҳокама қилайлик.

* + 1. **Нима учун ундан фойдаланиш керак**

Умид қиламизки шу пайтгача сиз Spring Framework ёрдамида амалга оширилши мумкин бўлган муаммоларни бошқа бир турдаги битта ёки бир нечта frameworkлар ёки бўлмаса APIлар ёрдамида ҳал қилган бўлишингиз мумкин. Шу боис ҳам сизда нима учун Spring билан ишлашни ўрганиш учун ўз қимматли бўлган вақтимни ажратишим керак деган савол туғилиши таъбий ҳол ҳисобланади. Spring ўз навбатида дастурий маҳсулотлар ишлаб чиқиш фаолияти билан шуғулланадиган дастурчилар учун иловаларни ишлаб чиқиш жараёнида дуч келиши мумкин бўлган мураккабликлар ва хатоликларга мойил бўлган кодлардан самарали изолация қиладиган компонент модели ва соддалаштирилган APIлар тўпламини тақдим этишдан ташқари яна бир нечта муҳим сабаблари мавжуд. Келинг улар билан бирма-бир танишиб чиқамиз, улар қуйидагилардан иборат:

***Сифатлилиги*** - Мазкур Spring Framework модуллар, пакетлар, синфлар тузилмаси ва APIларни умумий лойиҳлашдан тортиб, токи уларни иловаларга тадбиқ қилиш ҳамда дастур кодини тест синовларидан ўтказишни ўз ичига олган, юқори сифатли очиқ кодли дастурий таъминот ҳисобланади.

***Модуллилиги*** – Юқорида таъкидлаб ўтганимиздек, мазкур Framework 20 га якин асосий модулларга ажратилган, бу эса ўз навбатида дастурчиларга ўз иловаларига бутун бир Frameworkни тўлалигича қўшиб олмасдан, балки ўзларига керакли модулларни танлаш имкониятини яратиб беради.

***Энг яхши амалиётлар тарғиботи*** – Spring ёрдамида илгари биз ўрганганимиз объектларинг содда модели(POJO)га асосланган дастурлаш асосида иловаларни ишлаб чиқишда компонент моделлари, модулли тестлаш синовларини амалга ошириш ва бошқа турдаги энг яхши амалиётларни тадбиқ этишни тарғиб қилади.

***Тез ва осон ўрганиш мумкинлиги*** –Spring таркибидаги APIларнинг содда ва ихчамлиги туфайли уни ўрганиш қийин эмас, балки уни осонлик билан тезда ўрганиш мумкин бўлади. Сиз тартиб билан Springни ўзраниш жараёнида мазкур тузилманинг умумий конуниятлари намаён бўлиб боради. Шу билан биргаликда юзлаб сондаги онлайн ва босма нашрлар, манбалар сизнинг ихтиёрингизда бўлади, шу жумладан асосий ишлаб чиқарувчилар тамонидан тақдим этиладиган хабарлар ва ягликлар ҳам жуда фойдали ҳисобланади.

***Машхурлиги*** – Мазкур Frameworkнинг қай даражада машхурлигини кўп сонли нашрлар, вебсайтлар ва бўш иш ўринлари ҳақидаги элонлар ҳажми ҳам тасдиқлаб беради, ҳозирги вақтда Spring деярли барча жойда мавжуд ва жуда ҳам кенг фойдаланиб келинмоқда.

Ушбу китобни ўрганиш давомида сиз учун манфаатли бўлган қўшимча билимларни эгаллашингиз учун жуда ҳам фойдали ҳисобланган қўлланмалардан бири, яъни Craig Walls муаллифлиги остида 2011-йилда Manning нашрийтида тамонидан чоп этилган «Spring in Action» китобининг учинчи нашрини ўқишни тавсия қиламиз.

Spring жуда кўп имкониятларни таклиф қилади, албатта унинг ландшафтини тушуниш ва англаш учун маълум бир вақт талаб этилади. Лекин шуни ишонч билан айтиш мумкинки мазкур ҳаракатлар бунга арзийди. Springни ўрганиш ва унинг ёрдамида турли ҳилдаги муаммоларни ҳал қилиш учун фойдаланиш орқали сиз турли технологияларни қандай қилиб бирлаштириш ва ўз иловларингза қўллашни ўрганиб оласиз. Ўзингиз учун керакли бўладиган конфигурация учун қатъий белгилаб қўйилган махсус параметрларни ўз синфларингиздан ташқарида ва марказлашган стандарт жойларда сақлашингиз мумкин бўлади. Шунингдек сиз интерфейсларга асосланган синфлар орасидаги боғланишларни лойиҳлаш орқали дастурий таъминотга қўйиладиган ўзгарувчан талабларни тўлақонли қондиришингиз мумкин бўлади. Ва ниҳоят, сиз камроқ ҳаракат ва қисқа муддат ичида кўп ҳажмдаги ишларни амалга оширасиз, чунки бу ерда барча маъсулиятни Spring Framework ўз зиммасига олади, сиз эса факатгина бизнес муаммоларни ҳал қилиш учун зарур бўлган кодларни ёзишга бор этиборингизни қаратишингиз талаб этилади.

Умид қиламизки энди сиз мазкур framework қандай имкониятлар тақдим этиши ҳақида умумий тасаввурга эга бўлдингиз. Келинг энди 1.2-расмда тасвирланган контейнернинг ядроси билан янада чуқурроқ танишиб чиқайлик. Springнинг асосий контейнери қолган барча модуллар учун инверсияси бошқарувини(IoC) ва DI мослашувчанлигини амалга ошириши имкониятини тақдим этади.



***1.2-расм. Контейнернинг асоси(ядроси)***

* 1. **Мослашувчан конфигурациялар**

Барчамизга сир эмаски IoC бундан бир неча йил илгари худди Springдаги каби DI контейнерлари орқали машҳурликка эришди. Гарчи бу интернет кўп йиллар олдин мавжуд бўлган бўлса ҳам, бу кўплаб сондаги дастурчилар учун янги ҳисобланган ҳамда нотаниш концепция хисобланади. Ушбу бўлимда биз сизларга нима эканлиги тушунтириб ўтамиз ҳамда унга таъсир этувчи кучларни кўриб чиқамиз. Шунингдек сиз ўз қўлларнигз ёрдамида бу ишларни амалга ошириб кўрасиз, бунинг натижасида эса Spring контейнерини қандай қилиб созлаш мумкинлигини кўриб чиқасиз.

* + 1. **Боғланишларни созлашнинг эски усули ҳақида**

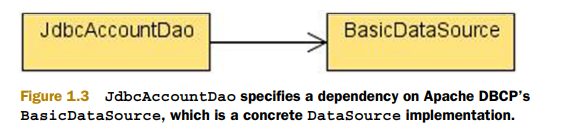
Келинг яхшиси буни тушуниш учун қўйида келитирлган дастур кодидаги наъмуналар орқали ундаги маълумотларга учун руҳсат олиш объекти(DAO) ва маълумотлар манбаи(DataSource) ўртасидаги боғланишларни кўриб чиқамиз. Бу ерда DAO билан DataSource биргаликда ишаши учун сиздан дастлаб JdbcAccountDao синфида DataSource яратишингиз ҳамда уни керакли параметрлар ёрдамида инициализация қилиш талаб этилади.

*// Source project: sip01, branch: 01 (Maven Project)*

**package** com.springinpractice.ch01.dao.jdbc;  
**import** org.apache.commons.dbcp.BasicDataSource;  
**import** com.springinpractice.ch01.dao.AccountDao;

**public class** JdbcAccountDao **implements** AccountDao {  
 **private** BasicDataSource **dataSource**;  
 **public** JdbcAccountDao() {  
 **dataSource** = **new** BasicDataSource();  
 **dataSource**.setDriverClassName(**"com.mysql.jdbc.Driver"**);  
 **dataSource**.setUrl(**"jdbc:mysql://localhost:3306/springbook"**);  
 **dataSource**.setUsername(**"root"**);  
 **dataSource**.setPassword(**""**);  
 }  
}

Мазкур код наъмунасида JdbcAccountDao синфи JDBC DataSource билан боғлиқлигини кўришимиз мумкин. Албатта бу ерда интерфейсларни ишлаб чиқиш(кодлаш) ва ундан фойдаланиш ҳеч қандай шубҳасиз энг яхши амалиёт ҳисобланади. Қуйидаги 1.3-расмда тасвирланганидек BasicDataSource учун зарур бўлган боғланиш яратилган, шунингдек DataSourceнинг тадбиқ этилиши учун эса Apache Commons Database Connection Pool(DBCP) лойиҳасидан фойдаланилганини кўришимиз мумкин бўлади.



***1.3-расм. JdbcAccountDao тамонидан Apache DBCPнинг BasicDataSource билан боғланишини белигилаб беради ҳамда муайяан бирDataSource ёрдамида тадбиқ этилган***

Бу ерда энг яққол кўзга куринадиган муаммо шундаки, JdbcAccountDao синфи маълумотлар манбаи(DataSource)га уланишни амалга ошириш учун керак бўлган барча конфигурациялар батафсил кўрсатиб ўтилганлигидир. Яна бир йирик муаммолардан биттаси кўплаб сондаги DAOлар манбага уланиш маълумотларини тақдим этишлари талаб қилинишидадир. Мазкур турдаги ёндашув натижасида DataSourceнинг бирон бир параметри ёки кофигурациялар ўзгарганида ҳар сафар дастур кодини ўзгартиришни, уни қайта компиляци қилишни ҳамда уни қайта жойлаштиришни талаб этади.

Бунинг олидини олиш учун сиз керакли параметрларни java.util.Properties ёрдамида амалга оширишингиз ва конфигурацияларни дастур кодидан ташқарига олиб ўтишингиз мумкин. Лекин шунга қарамасдан у барибир сизда нозик муаммо сифатида қолаверади. Олдинги дастур коди наъмунасида кўрсатилган синфнинг ўзи боғланишни белгилайди ҳамда уни бошқаради. Келинг энди мазкур бошқарувларни боғланишлар инецияси ёрдамида қандай амалга ошириш мумкинлигини қўриб чиқамиз.

* + 1. **Боғланиш инекцияси ҳақида**

Аниқ бир BasicDataSourceга боғлиқ бўлишни бартараф этишнинг усулларидан бири бу ташқи боғланишни амалга ошириш ҳисобланади, бунинг учун эса JdbcAccountDao синфига DataSource сифатидаги боғлиқликни амалга ошириш лозим бўлади. Будай юқори мослашувчанлик эса сизга жуда ҳам кенг қўлайлиқлар яратади чунки сиз конфигурацияларни факатгина битта жойда осонгина ўзгартиришингиз мумкик бўлади. Агарда сиз DataSourceни инекция қилишдан олдин уни проксиламоқчи бўлсангиз, уни ҳам амалга оширишингиз мумкин. Дастурни тестлаш ишларини амалга ошириш мақсадида mock маълумотлар манбасидан фойдаланишга тўғри келадиган бўлса, уни ҳам амалга оширишингиз мумкин. Шунингдек мазкур компонентнинг конструкторлари боғланишни амалга оширишга имкон берадиган таркибий қисмларга таянадиган бўлса, у холда DI сизга илгари мавжуд бўлмаган бир қанча қулайликларга эга бўлган мослашувчанликни тақдим этади.

Шуни яхши англаб етишимиз лозимки DI ишлаши учун DataSourceни алоҳида яратишингиз лозим бўлади, кейинчалик эса унинг ёрдамида DAO яратишингиз ёки қўйида кўрсатилгандек унинг методлари ёрдамида DAOни ўрнатишингиз керак бўлар экан.

*// Source project: sip01, branch: 02 (Maven Project)*

**package** com.springinpractice.ch01.dao.jdbc;  
**import** javax.sql.DataSource;  
**import** com.springinpractice.ch01.dao.AccountDao;  
**public class** JdbcAccountDao **implements** AccountDao {  
 **private** DataSource **dataSource**;  
 **public** JdbcAccountDao() {}  
 **public void** setDataSource(DataSource dataSource) {  
 **this**.**dataSource** = dataSource;  
 }  
}

Юқоридаги дастур кодига этибор қаратадиган бўлсак DAO энди BasicDatasource билан қаътий боғланишга эга эмас. Натижада BasicDataSourceни импорт қилиш олиб ташланганини сезасиз. Боғланишлар setter методи ёрдамида амалга оиширилгани учун DataSourceга уланиш учун эндиликда констурктор учун зарурият сезилмайди. Мазкур кодни refactoring қилиш асосида ана бир ёндошув вужудга келади, яъни DataSource setterдан фойдаланиш ўрнига конструкторнинг аргументи сифатида тақдим этиш ҳисобланади. Ҳар иккала ҳолатда ҳам коднинг яхшиланишини кузатишимиз мумкин. Ўз навбатида сиз ушбу боғланишларни коднинг бошқа қисмларига жойлаштиришингиз мумкин ва бу ҳақида аосоли тортишишингиз мумкин. Куйида DAOни яратиш учун мўлжалланган serviceга каранг:

*//Source project: sip01, branch: 02 (Maven Project)***package** com.springinpractice.ch01.service;  
**import** java.util.Properties;  
**import** java.io.InputStream;  
**import** org.apache.commons.dbcp.BasicDataSource;  
**import** com.springinpractice.ch01.dao.jdbc.JdbcAccountDao;

**public class** AccountService {  
 **private** JdbcAccountDao **accountDao**;

**public** AccountService() {  
 **try** {  
 Properties props = **new** Properties();

InputStream inputStream = **this**.getClass()

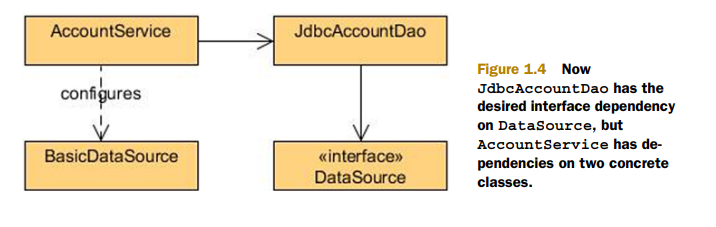
.getClassLoader()  
 .getResourceAsStream(**"dataSource.properties"**);

props.load(inputStream);

BasicDataSource dataSource = **new** BasicDataSource();  
 dataSource.setDriverClassName(  
 props.getProperty(**"driverClassName"**));  
 dataSource.setUrl(props.getProperty(**"url"**));  
 dataSource.setUsername(props.getProperty(**"username"**));  
 dataSource.setPassword(props.getProperty(**"password"**));

**accountDao** = **new** JdbcAccountDao();  
 **accountDao**.setDataSource(dataSource);  
 } **catch** (Exception e) {  
 **throw new** RuntimeException(e);  
 }  
 }  
}

Бир жиҳатдан олиб қаранганда, бу ерда вазият янада мураккаблашганини кўришимиз мумкин. Бу ерда биз худди кераксиздек бўлган AccountService ва BasicDataSource орасида боғланишни амалга оширдик. Шунингдек бизда JdbcAccountDao ва JdbcAccountDao ўрасида ҳам боғланиш мавжуд, шунинг учун ҳам биз ўзимизни ҳамон ўзимиз ҳаракатга келирган кайиқнинг ичида ўтиргандек ҳис қиламиз.



***1.4-расм***

Юқорида келтирилган 1.4-расмга қарайдиган бўлсак муайян бир тизим учун барча боғланишлари графигини мураккаб ва мослунвчан бўлмаган тугунлар орқали ўзгартишиш мурккаб эканни англашимиз осон бўлади.

Лекин шуни бир нарсани унитмаслигимиз лозим бўладики бу ерда келтирилган мисолда DI қўлланилиши муваффақиятсиз тажриба эканлигини англатмайди, балки биз уни тўғри талқин қилишимиз учун кенг йўл очиб беради. Мазкур ишларни тўғри йўлга қўйишимиз учун инекция нима вазифа бажараётганини қайт кўриб чиқишимиз керак.

* + 1. **Бошқарув инверцияси**

Келинг энди DI ни мижоз кодидан узоклаштирамиз ва уни Springга олиб ўтамиз, ушбу ценарий ёрдамида мижоз коди AccountServiceни талаб қилмайди ва уни қидирмайди. Бунинг ўрнига мижоз коди ишга тушиш жараёнида яъни иницализация килинаётганида AccountService шаффоф тарзда боғланади. Қуйидаги код ёрдамида AccountDao интерфеси ёрдамида AccountDao билан катъий боғланиш тасвирланган.

*//Source project: sip01, branch: 03 (Maven Project)***package** com.springinpractice.ch01.service;  
**import** com.springinpractice.ch01.dao.AccountDao;  
  
**public class** AccountService {  
 **private** AccountDao **accountDao**;  
 **public** AccountService() {}  
 **public void** setAccountDao(AccountDao accountDao) {  
 **this**.**accountDao** = accountDao;  
 }  
}

Бу ердаги боғланиш занжирини қандай аниқлаш мумкин? Бу занжирни аниқлаш учун Spring дан фойдаланим амалга шириш сулларидан бири ҳисобланган XML асосида декларатив усулда қўийда кўрсатилгандек йўлга қўйиш мумкин. Қўийдаги 1.1- Listingга этибор қаратишингизни сўраймиз.

Listing 1.1 Spring configuration file that specifies object relationships

*<?***xml version="1.0" encoding="UTF-8"***?>  
<!-- Source project: sip01, branch: 03 (Maven Project) -->*<**beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans-3.1.xsd"**>  
 <**bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"  
 destroy-method="close"**>  
 <**property name="driverClassName" value="com.mysql.jdbc.Driver"**/>  
 <**property name="url" value="jdbc:mysql://localhost:3306/springbook"**/>  
 <**property name="username" value="someusername"**/>  
 <**property name="password" value="somepassword"**/>  
 </**bean**>

<**bean id="accountDao"  
 class="com.springinpractice.ch01.dao.jdbc.JdbcAccountDao"**>  
 <**property name="dataSource" ref="dataSource"**/>  
 </**bean**>  
 <**bean id="accountService"  
 class="com.springinpractice.ch01.service.AccountService"**>  
 <**property name="accountDao" ref="accountDao"**/>  
 </**bean**>  
</**beans**>

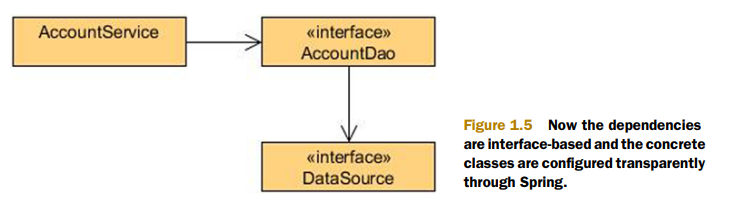
Агарда сиз эндигина Spring билан ишлашни бошлаган бўлсангиз, у ҳолда мазкур конфигурация сиз учун нотаниш бўлиши мумкин, аммо унинг маъноси сиз учун етарлича аниқ бўлишига шубҳамиз йўқ. Демак буни ҳам изоҳлаб ўтамиз

*1-қисмда биз DataSourceни элон қиламиз ва унинг конфигурацияси учун зарур бўладиган параметрларни ўрнатиб оламиз*

*2- қисмда эса биз JdbcAccountDao ни элон қиламиз ва уни биз учун керакли бўлган dataSource ўртасидаги боғланишни амалга оширамиз*

*3- қисмда эса худди юқоридаги каби AccountServiceни элон қиламиз ҳамда уни JdbcAccountDao билан боғланишини таъминлаймиз*

Якуний натижа шундан иборат бўладики, бу ерда сервис мавжуд барча боғланишларни ўз зиммасига олади ва ўз нвабатида юқоридаги конфигурация мазкур сервис учун тўлиқлигича шаффоф хисобланади. Барча муносабатлар тўғри йўлга кўйилган янги кўринишдаги боғланишлар 1.5-расмда синфлар диаграммаси шаклида келтирилган.



***1.5-расм***

Кўриб турганингиздек бу ерда синф тузилмаси бир нечта қатламлардан ташкил топган бўлиб, яъни кўп қатламли ёндашув асосида ишлаб чиқилган. Олдинги 1.4-расмдан фарқли ўлароқ мазкур расмда боғланишлар ўзидан қўйи пағонада турган қатламларга боғлиқ эканига аҳамият беришмиз мумкин бўлади, шунингдек мазкур боғланишларнинг барчаси интерфейс сифатида ифода этилган. Бу эса бизга боғланишлар графигини соддалаштириш имконини беради керакли тугунларни бошаси билан алматиришни осонлаштиради. Кейинги бўлимда биз сизларга ўз боғланишларингизни Spring ёрдамида қандай қилиб амалга ошириш ва уни бошқаришни кўрсатиб берадиган мисолдан фойдаланамиз.

* 1. **Содда турдаги Bean конфигурациялар ҳақида**

Биз сизлар билан биргаликда олдинги саҳифаларимизда наъмуна сифатида келтирилган мисоллар ёрдамида кичик ҳажмдаги иловаларда Spring Framework орқали DIни қандай қўллаш мумкинлигини кўриб чиқдик. Куйида келтирилган илова аслида ҳеч кандай жиддий муаммоларга дуч келмайсиз, лекин Spring ёрдамида боғлиқликларни ифодалаш ва уларни бошқариш асосларини етарли даражада ўрганиб олишингиз мумкин бўлади. Бу ерда сиз дастлаб домен объектни яртишнингиз лозим бўлади кейинчалик CSV форматидаги файлдан ўқиш учун махсус DAO ишлаб чиқасиз, шунингдек service хизматини ҳам яратсиз ва ниҳоят уларнинг барчасини бир бирига боғлашни ўрганиб оласиз.

Мазкур иловада комунал соҳада фаолият олиб борувчи ташкилотнинг мавжуд қарздорликлари рўйҳатини шакллантиришга эътибор қаратилган. Шу босидан ҳам сиз бу ерда 30 кун ва ундан ортиқ муддат муддат давомида тўланмаган ҳисоб рақами сифатида қаралади. Бу ерда service хизмати муддати ўтган хисоб рақамларини аниқлашни ўз зиммасига олади, лекин танлаб олинган ташкилотнинг хисоб рақамлари рўйхатини олиш учун DAOга ваколат берилиши талаб этилади. Шуни унитмаслик лозимки DAOни яратишдан олдиз сиз домен объектини аллақачон яратиб олган бўлишингиз керак бўлади.

* + 1. **Ҳисоб рақами учун домен объектини яратиш ҳақида**

Қуйидаги листинг ёрдамида Account дем номланувчи домен объектини яратамиз. Мазкур объект кейинчалик илованинг бошқа қисмларида фойдаланиш учун зарур бўладиган ва Diни ифодалаш учун зарур бўлган майдонлар ва усуллардан ташкил топган.

Listing 1.2 Basic account bean, Account.java

*// Source project: sip01, branch: 04 (Maven Project)*

**package** com.springinpractice.ch01.model;

**import** java.math.BigDecimal;  
**import** java.util.Date;  
  
**public class** Account {  
 **private** String **accountNo**;  
 **private** BigDecimal **balance**;  
 **private** Date **lastPaidOn**;  
  
 **public** Account(String accountNo, BigDecimal balance, Date lastPaidOn) {  
 **this**.**accountNo** = accountNo;  
 **this**.**balance** = balance;  
 **this**.**lastPaidOn** = lastPaidOn;  
 }  
  
 **public** String getAccountNo() {  
 **return accountNo**;  
 }  
  
 **public** BigDecimal getBalance() {  
 **return balance**;  
 }  
  
 **public** Date getLastPaidOn() {  
 **return lastPaidOn**;  
 }  
}

Шу ўринда ьитта нарсани эслаб ўтиш ўринли бўлади яъни реал ҳолатдаги иловаларда хисоб рақамларидаги маълумотлар хавода ўз-ўзидан пайдо бўлиб қолмайди. Сиз мазкур маълумотларни аниқ бир маълумотлар базалари, файллар ёки бошқа турдаги маълумотлар сақлаш ва уни ўқиш имкониятига эга бўлган тизимлар ёрдамидан фойдаланишингиз лозим бўлади. Мисол учун сиз бир-бири билан вергу орқали ажратилган қийматларни қўйида кўрсатилган маълумотлар билан тўлдирилган CSV форматидаги файлни таҳлил қилиш учун мўлжалланган DAO яратишингиз лозим бўлади.

100,0,09012008  
200,100,08012008  
300,-100,09012008

Ушбу accounts.csv файл таркибидаги биринчи майдон хисоб рақами, иккинчи майдон баланс (мусбат ёки манфий) ва ниҳоят учинчи майдонда охирги тўланган сана MMDDYYYY каби форматда ўз ифодасини топган. Олдинги 1.2-бўлимда таъкидлаб ўтилганидек

* + 1. **Ҳисоб рақами учун DAO интерфейсини яратиш ва уни жорий қилиш ҳақида**
    2. **Spring ёрдамида CsvAccountDaoни созлаш ҳақида**
    3. **Ҳисоб рақамлари бўйича мавжуд қарздорликларни аниқлаш ҳизматини ишлаб чиқиш**
    4. **AccountService ҳамда CsvAccountDao орасидаги боғланишларни амалга ошириш ҳақида**
  1. **Bean компонентлар оарсида боғланишларyb XML ёрдамида амалга ошириш ҳақида**
     1. **Beanларнинг номланиш майдони ҳақида умумий тушунчалар**
     2. **Beanларнинг амал қилиш соҳалари ҳақида**
     3. **P типдаги ном майдони ҳақида**
     4. **С типдаги ном майдони ҳақида**
  2. **Аннотациялар ёрдамида компанентларни сканерлаш ва автоматик боғлаш ҳақида**
     1. **@Autowired ҳақида**
     2. **Стереотип аннотациялар ҳақида**
     3. **Компанентларни сканерлаш**
     4. **XML ва аннотацияларнинг афзалликлари**
  3. **Боб бўйича хулоса**

Ушбу бобда Spring Framework ҳақида қичқача маълумотлар тақдим этилган бўлиб, Springнинг энг асосий қисмлари ва тамойиллари ёритиб берилган, жумладан бошқарув инверсияси ва боғланишларни амалга ошириш усуллари баён этилган. Сиз азизларга тақдим этилган бир нечта мисоллар орқали иловалар таркибидаги боғликликларни бошқаришда Springдан фойдаланишнинг ўзига хос афзалликларини ҳам кўрсатиб беришга ҳаракат қилдик, боб якунида эса ушбу боғлиқликларни амалга оширишда XML ва Java аннотациялар ёрдамида амалга ошириш усулларини ҳам тасвирлаб беилди.

Кейинги бобда энди контейнерлардан бироз узоқлашган ҳолда ва бизлар учун жудаям фойдали бўлган бир нечта компонентларни ўрганамиз, яъни Spring таркибида мавжуд бўлган persistence алоқларини йўлга қўйиш, объектнинг боғланишлар харитасини ишлаб чиқиш ва транзакцияларни бошқариш каби муҳим имкониятларини кўриб чиқамиз. Мазкур бобда келтириладиган мисолларда ишлатилган компонентлардан сиз кейинчалик ўз иловаларингизда кенгроқ фойдаланиш имкониятига эга бўласиз.