

Дәріс 5. Labview Кейс (Case) және Sequence Structure құрылымдары

Мақсаты: Студенттерді **Enum** (таңдау тізімі) деректер типімен, **Case Structure** (шарттық тармақтау құрылымы) және **Sequence Structure** (реттік орындалу құрылымы) элементтерімен таныстыру. Олардың көмегімен бағдарлама логикасын басқару, әртүрлі жағдайларды өңдеу және реттілікпен орындалатын әрекеттерді ұйымдастыру жолдарын үйрету.

Кіріспе.

Бұл лекцияда біз LabVIEW-дің **Enum** (таңдалатын мәндер жиыны), **Case Structure** (шарттық тармақтау) және **Sequence Structure** (реттік орындалу) сияқты негізгі басқару құрылымдарымен танысамыз. Бұл құрылымдар бағдарлама логикасын нақты және тиімді құруға мүмкіндік береді.

- **Enum** – бағдарламаны жеңіл оқуға және қолданушы интерфейсін қарапайым етуге мүмкіндік беретін, сандық мәндерге атау берілген деректер типі.
- **Case Structure** – белгілі бір шартқа байланысты түрлі код блоктарын орындауға арналған құрылым.
- **Sequence Structure** – әрекеттердің нақты тәртіппен орындалуын қамтамасыз етеді.

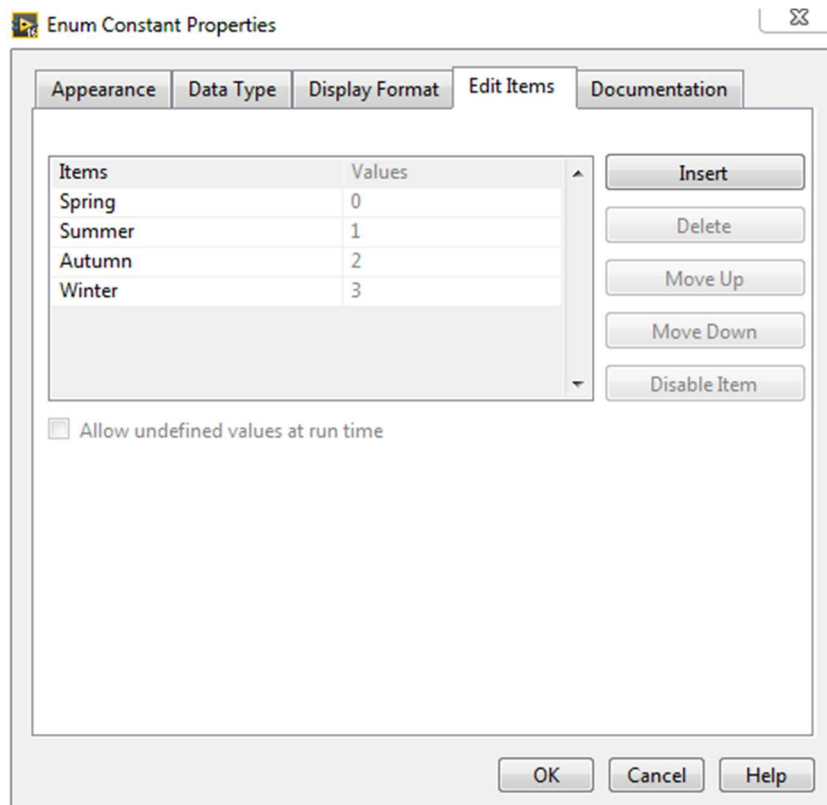
Осы басқару құрылымдарын меңгеру арқылы студенттер **шартқа тәуелді логикаларды** және **кезең-кезеңімен орындалатын процестерді** LabVIEW ортасында сауатты түрде ұйымдастыруды үйренеді.

Санақталған түр ([enum](#)) (Сурет 1) – сәйкес бүтін мәндері бар жол белгілерінің тізімі. Ол LabVIEW ішінде таңбасыз бүтін сан ретінде өңделеді (U8 , U16 - әдепкі және U32).

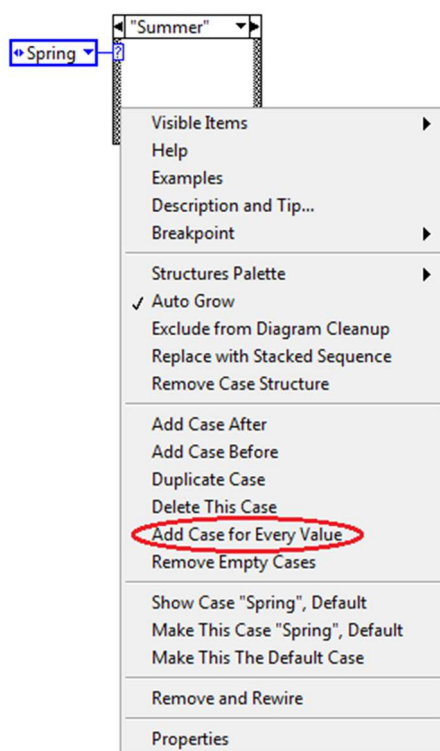
Мысалы , бізде төрт ауа райы маусымын өткізуге арналған санау болуы мүмкін , бұл жағдайда әрқайсысы үшін сандық мәнге ие боламыз: 0-" көктем " , 1-" жаз " , 2-" күз " және 3-" қыс " .

Жағдай құрылымының (Case Structure) регистрді таңдау құралына нөмірді жалғаған кезде , нөмірдің әрбір мәні үшін әртүрлі жағдайларды жасай аласыз. Сандық мәндердің әрқайсысы үшін регистр жасау үшін регистр құрылымы жақтауын тінтуірдің оң жақ түймешігімен басып, **әрбір мән үшін Іс қосу пәрменін таңдаңыз.**

Мысалы , егер тізімде төрт ауа райы маусымы болса, сізде "көктем" , "жаз" , "күз" және "қыс" әр мән үшін төрт түрлі жағдай болуы мүмкін .

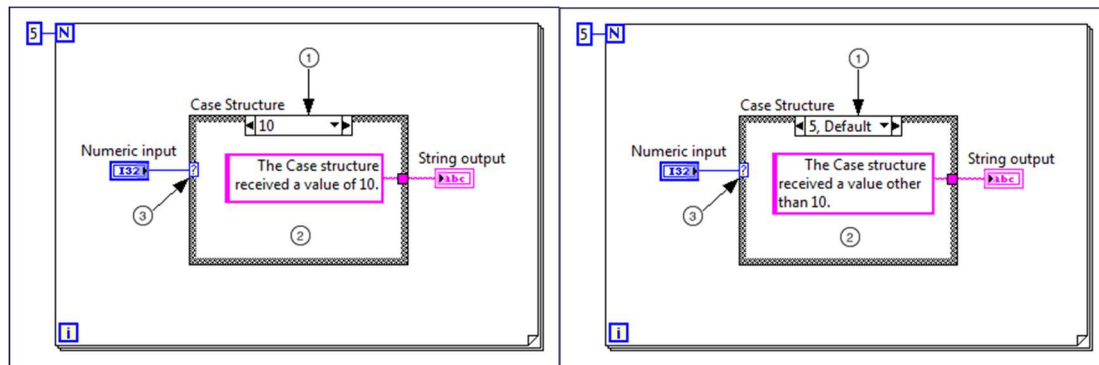


Сурет 1 - Санақталған түр (enum) интерфейсі



Сурет 2 - Case Structure құру

Case Structure құрамдас бөліктері



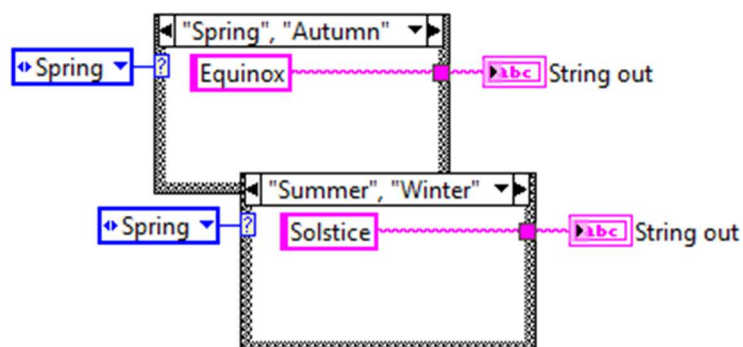
Сурет 3 - Case Structure құрамдас бөліктері

Сурет 3 -те көрсетілген белгілер:

- ① **Таңдау белгісі** — Байланысты іс орындалатын мән(дер)ді көрсетеді. Бір мәнді немесе **мәндер ауқымын** көрсетуге болады . Сондай-ақ , **әдепкі жағдайды көрсету** үшін селектор белгісін пайдалануға болады .
- ② **Ішкі диаграмма(регистр)** — Регистрді таңдау құралына жалғанған мән селектор белгісінде пайда болатын мәнге сәйкес келгенде орындалатын кодты қамтиды. Ішкі диаграммалардың санын немесе ретін өзгерту үшін Case құрылымының жиегін тінтуірдің оң жақ түймешігімен басып, сәйкес опцияны таңдаңыз.
- ③ **Іс таңдау құралы** — Енгізілген деректердің мәніне қарай орындалатын істі таңдайды. Енгізілетін деректер логикалық, жол, бүтін сан, нөмірленген түр немесе **қате кластері** болуы мүмкін . Іс таңдау құралына жалғайтын деректер түрі селектор белгісіне енгізуге болатын рұқсат етілген жағдайларды анықтайды.

Сонымен қатар, сізде бір жағдайда бірнеше санау элементтері болуы мүмкін. Таңдаушы белгі жағдайында нөмірдің жол мәндері **қос** тырнақшадағы жолдар ретінде көрсетіледі. Бір регистрге бірнеше элемент қосқыңыз келсе, олардың аттарын үтірмен бөлінген тырнақшаға қосуға болады.

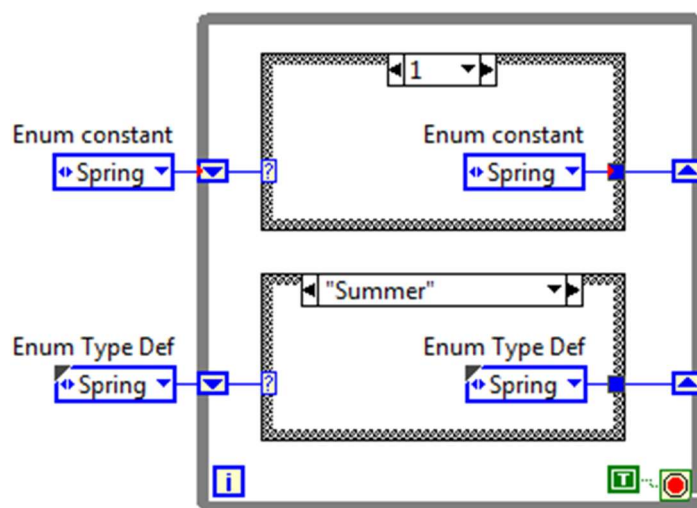
Мысалы , егер біз күн мен түннің теңелуі үшін Кейс құрылымымыздың көктем мен күзді бірге өңдеуін қаласақ; және Жаз бен Қыс бірге күн тоқырауы болғандықтан, бізде: 1-жағдай – «Көктем», «Күз» . 2 жағдай – «Жаз», «Қыс» .



Сурет 4 - Case Structure қолдану мысалы

Қосымша ақпарат

- Санақталған деректер түрі (Enum) осы мақалада қарастырылмаған сақина деректер түрінен ерекшеленеді. Оның айырмашылықтары мен қолданылуы туралы қосымша ақпарат алу үшін .
- Санақталған түрлерді (сандарды) пайдаланған кезде, басқару элементінің түр анықтамасын жасау ең жақсы тәжірибе болып табылады. [Түр анықтамаларын жасау](#) нөмірге элементті қосқан немесе жойған сайын кодты қайта жазу қажеттілігінен сақтайды. Осылайша, олардың біреуін өзгерткен кезде, Іс құрылымы белгілеріндегі сандармен ауыстырылған жол мәндерін алмайсыз. **Ескертпе :** Төмендегі 5-суреттегі жоғарғы Іс құрылымы жол белгілерінің орнына сандарды көрсетеді, себебі бір нөмір тұрақтысы оның мәндерінің бірінде өзгертілген. Сондықтан LabVIEW барлық іс белгілерін сандық түрге мәжбүрлейді және Іс құрылымын тек сандық мәндерді күтуге мәжбүр етеді.



Сурет 5 - Enum тұрақтысын қолдану арқылы жасалған Case құрылымы

Кезеңдік құрылымдар (Sequence Structure): код бөлімдерін ретімен орындау

Тізбек құрылымында бірізділікпен орындалатын бір немесе бірнеше [ішкі диаграммалар](#) немесе фреймдер бар. Блок-схеманың қалған бөліктеріндегідей реттілік құрылымының әрбір жақтауында деректерге тәуелділік түйіндердің орындалу ретін анықтайды.

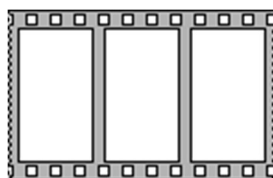
Реттік құрылымдардың екі түрі бар — [Flat Sequence](#) құрылымы және [Stacked Sequence](#) құрылымы. Тізбек құрылымдарын үнемді пайдаланыңыз, себебі олар кодты жасырады. Орындау ретін басқару үшін реттілік құрылымдарына емес, деректер ағынына сеніңіз. Жүйелік құрылымдармен қатар жергілікті айнымалы мәнді пайдаланған кезде деректер ағынының солдан оңға қарай парадигмасын бұзасыз.

Деректер ағынын басқаруға көмектесу үшін оның орнына қате кластерлерін пайдалануға болады. Егер **ағын параметрлері** қол жетімді болмаса және VI ішінде реттілік құрылымын пайдалану қажет болса, Flat Sequence құрылымын пайдалануды қарастырыңыз.

Реттік құрылымның шығыс туннельдерінде **Case** құрылымдарынан айырмашылығы тек бір дерек көзі болуы мүмкін. Шығару кез келген кадрдан шығарылуы мүмкін. Case құрылымдарындағы сияқты, кіріс туннельдеріндегі деректер Flat Sequence немесе Stacked Sequence құрылымындағы барлық кадрлар үшін қолжетімді.

Жазық реттілік (Flat Sequence) құрылымы

Төменде көрсетілгендей Flat Sequence (Сурет 6) құрылымы кадрларды солдан оңға қарай және кадрға жалғанған барлық деректер мәндері қолжетімді болғанда орындайды. Фрейм орындауды аяқтаған кезде деректер әрбір кадрды қалдырады. Бұл бір кадрдың кірісі басқа кадрдың шығысына байланысты болуы мүмкін дегенді білдіреді.



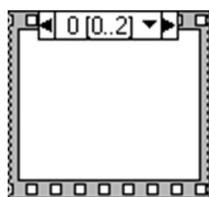
Сурет 6 - Flat Sequence

Flat Sequence құрылымында **кадрларды қосқанда немесе жойған** кезде құрылым автоматты түрде өлшемін өзгертеді.

Тегіс ретті жинақталған реттілікке өзгертіп , содан кейін тегіс реттілікке қайта оралсаңыз, LabVIEW барлық кіріс терминалдарын тізбектің бірінші кадрына жылжытады. Соңғы Flat Sequence Stacked Sequence сияқты жұмыс істейді. Бірінші кадрдағы барлық кіріс терминалдары бар жинақталған тізбекті тегіс реттілікке өзгерткеннен кейін, сымдарды бастапқы тегіс реттілікте орналасқан жерге жылжытуға болады.

Стектелген реттілік құрылымы (Stacked Sequence)

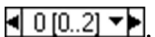
Төменде көрсетілген Stacked Sequence (Сурет 7) құрылымы әрбір кадрды жинақтайды, осылайша сіз бір уақытта тек бір кадрды көресіз және 0 кадрды, содан кейін 1 кадрды және соңғы кадр орындалғанша осылай жалғаса береді .



Сурет 7 - Stacked Sequence

Stacked Sequence құрылымы деректерді соңғы кадр орындалғаннан кейін ғана қайтарады. Блок-схемада бос орынды сақтағыңыз келсе, Stacked Sequence құрылымын пайдаланыңыз.

Flat Sequence құрылымындағы кадрлар арасында деректерді беруден айырмашылығы, Stacked Sequence құрылымында деректерді кадрдан кадрға беру үшін жүйелілік жергілікті мәндерін пайдалану қажет.

Стектелген рет құрылымының жоғарғы жағында келесідей көрсетілген реттілік селекторының идентификаторы ағымдағы кадр нөмірін және кадрлар ауқымын қамтиды .

Қол жетімді кадрларды шарлау және кадрларды қайта реттеу үшін реттілік селекторының идентификаторын пайдаланыңыз. Stacked Sequence құрылымындағы жақтау белгісі Case құрылымының регистр таңдау белгісіне ұқсас. Жақтау жапсырмасы ортасында жақтау нөмірін және әр жағында азайту және арттыру көрсеткілерін қамтиды.

Жақтау белгісіне мәндерді енгізу мүмкін емес. Stacked Sequence құрылымында кадрларды қосқанда, жойғанда немесе қайта реттегенде, LabVIEW кадр белгілеріндегі сандарды автоматты түрде реттейді.

Тапсырмалар

Тапсырма 1: Калькулятор логикасы (Enum + Case Structure)

Тапсырма 2: Қадаммен орындалатын жүйе (Sequence Structure)

Тапсырма 3: Бірнеше мәнді бір Case-те өңдеу (Advanced Enum + Case)

Бақылау сұрақтары

1. Enum басқару элементінің негізгі қызметін түсіндіріңіз.
2. Кейс құрылымының (Case Structure) жұмыс істеу принципін түсіндіріңіз.
3. Бір Case терезесіне бірнеше мәнді біріктірудің жолын түсіндіріңіз.
4. Sequence Structure дегеніміз не екенін түсіндіріңіз.
5. Duplicate Case ұғымын түсіндіріңіз.