«Бекітемін»

«Кәсіптік оқыту» ПЦК \_\_\_\_\_\_\_\_\_Б.Жадыра

«\_\_\_\_»\_\_\_\_\_\_2021ж

**«Түлкібас агробизнес және саяхат колледжі» МКҚК**

**№ 36 сабақ жоспары**

**Конструкторлар мен деконструкторлар.**

(сабақ тақырыбы)

**Модуль /пән атауы** Өндірістік оқыту

**Дайындаған педагог** Ускенбаева А.Н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(қолы)

2021 жылғы "\_\_\_\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_

**1.Жалпы мәліметтер**

     ІІ Курс, 2021 – 2022 оқу жылы, ЕТ-021 топ

**Сабақ типі** Білімді,білікті,іскерлікті дамыту,өндірістік оқыту.

**2. Мақсаты, міндеттер**

**Білімділік:** Контейнерлі кластар, алгоритмдер және итераторлар кіретін стандартты кітапханасының бөлігі, үлгілердің стандартты кітапханасы. Әрекеттер аймақтары мен атаулар кеңістіктерін ашып көрсету. Класс ұғымын толық түсіндіру. «Клиент-сервер» технологиясы бойынша класс әдістерін анықтау. Конструктор мағынасы. Деструктор мағынасы. Есептерді жеке кішігірім есептерге бөлу арқылышешудің әдістері мен тәсілдерін игеру  және қосалқы программаларды құрастыруға машықтану.

**Дамытушылық:** С++ тілінің стандартты кітапханасындағы енгізушығару ағындары кластарын түсіндіру. Жұмыс барысында өзін ісін сауатты, ұқыпты орындайтын жеке тұлғаны дамыту.

**Тәрбиелік:** Өздігінен ізденуге тәрбиелеу,танымдық іс-әрекетке баулу.

2.1 Оқу сабақтары барысында білім алушылар игеретін кәсіби біліктердің тізбесі

Студенттерге пәнді оқыта отырып, қосалқы программаларды құрастыруға машықтану, студенттің ойлау қабілетін дамыту.

**3. Сабақты жабдықтау**

      3.1 Оқу-әдістемелік құрал-жабдықтар, анықтамалық әдебиеттер

      3.2 Техникалық құралдар, материалдар

Интербелсенді тақта

**4. Сабақтың барысы**

**Сабақты ұйымдастыру**:

1. Студенттерді түгелдеу, дәрісхананың тазалығын тексеру.

2. Студенттер зейінін сабаққа аудару.

**Өткен тақырыптарға шолу:**

Ауызша сұрақтар қою:

1. ДБ пайдаланатын қосымшаларды құру және эксплуатациялау үшін қандай құралдар Delphi құрамына кіреді және олар қандай функцияны орындайды?

2. ДБ-мен жұмыс істеу үшін Delphi көмегімен құрылған программалардың тән ерекшелігі не болып табылады?

3. Қандай ДБ типтері Delphi-де қолданылады және олар нені білдіреді?

**Конструкторлар мен деконструкторлар.**

Конструкторлар мен деструкторлар объектінің жадыда болуы үшін жауап береді, яғни. сыныптың данасы үшін жадыны бөліп, оны босатыңыз.

Конструктор - бұл сыныпқа бекітілген ішкі бағдарламаның ерекше түрі. Оның мақсаты - сынып өкілдерін (даналарын) құру. Ол сыныптың жаңадан құрылған данасына сілтемені қайтаратын функция сияқты әрекет етеді, яғни. объектіге. Сонымен бірге жад класс данасының өрістерінің мәндерін сақтауға бөлінеді.

Деструктор - бұл сыныпқа бекітілген ерекше бағдарламаның түрі. Оның мақсаты - класстың данасын жою, яғни. дана үшін бөлінген жадыны босату және объект.

Конструкторлар мен деструкторларды жариялауға арналған синтаксис:

Түрі

<сынып атауы> = Сынып [{Ата-аналық сынып атауы>)]

... ... ...

Конструктор Конструктор атауы> [(<параметрлер>)]; [Қайтару;]

Деструктор <деструктор атауы> [(<параметрлер>)}; [Қайтару;]

Соңы;

Ескертулер:

• Конструкторлар мен деструкторлар әдетте сыныптың Public бөлімінде жарияланады.

• Сыныпта бірнеше конструкторларды жариялауға болады, бірақ көбінесе бір конструктор болады. Жалғыз конструктордың жалпы атауы.

• Бір сыныпта бірнеше деструкторларды жариялауға болады, бірақ көбінесе параметрлері жоқ бір деструктор болады (әрдайым!) Жою деп аталады.

• Destroy деп аталатын деструктордың декларациясын Override кілт сөзі жалғастыруы керек, егер әдепкі әрекет объектінің данасын жоюға мүмкіндік берсе, егер оны жасау кезінде қате пайда болса. Шындығында, қайта анықтау ата-баба әдісін жоққа шығарады

• Free әдісі сонымен қатар сыныптың даналарын жояды (жояды), оларды Nil үшін алдын ала тексереді.

Жүзеге асырушы құрылысшылар

Конструктордың міндеті - класстың данасын құру және оның денесінде орналасқан операторларды орындау. Конструктор ішіндегі кодтың мақсаты - жаңадан құрылған объект данасын инициализациялау. Конструкторды іске асырудың синтаксисі:

Конструктор <сынып аты>. <Конструктор аты> [(<параметрлер>}};

[<жарнама блогы>}

Баста

<Орындалатын операторлар>

Соңы;

Тұқым қуалаушы конструкторларды енгізу.

Конструктор <сынып аты>. <Конструктор атауы> [(<параметрлер>)];

[<жарнама блогы>}

Баста

Мұраланған <конструктор атауы> [(<параметрлер>)};

<тапсырыс бойынша өрістерді инициализациялау>

Соңы;

Ескертулер:

• Конструктордағы әр өріс үшін тағайындау операторы берілгеніне және барлық өрістер анықталмаған күйден әдепкі бойынша берілгенге ауысатынына көз жеткізіңіз, бірақ аудармашының өзі нөлдік мәнге дейін инициализациялайтыны белгілі («, Nil, False - басқа өрістер үшін). Классикалық емес өрістер үшін мұндай инициализация өте қолайлы, өйткені бұл оларды болашақта пайдалануға мүмкіндік береді.

• Мұраланған конструкторды шақыру үшін, қайта берілген әдісті қол жетімді ететін мұрагерлік кілт сөзін қолданыңыз. Оператордың күші - ескіні шақырып, содан кейін жаңасына оралуында.

• Жалпы, сіз бірінші орындаған операторға тиісті мұрагерлік конструкторды шақыруыңыз керек.

• Тек қана тапсырыс бойынша класта жаңа өрістер болмаса ғана, сіз оған конструктор жасай алмайсыз.

• Конструктор декларацияларында қайтару типі көрсетілмеген және процедура декларацияларына ұқсас болғанымен, конструкторлар процедураларға қарағанда функцияларға көбірек қолданылады. Конструктор - бұл жасырын функция деп айтуға болады - ол шақырылған кезде қолданылған кластың жаңа өкілін қайтарады. Қайтарылатын нәтиженің түрін нақты орнатудың қажеті жоқ, өйткені бұл тип қоңырау кезінде белгілі болған - бұл конструкторға шақыруда қолданылатын класс типі.

• Конструктор ішінде қайтарылатын мәннің нақты тағайындалуы жоқ. Бұл мүмкін, өйткені ол әрдайым сыныптың жаңадан құрылған және инициализацияланған данасына сілтеме (адрес) қайтарады.

• TComponent сыныбынан бастап Create конструкторы виртуалды болды және оны қайта анықтаған кезде Override директивалық сөзін көрсету керек.

Деструкторларды жүзеге асыру

Деструктор оны шақырған кезде қолданылған кластың данасын жояды, конструктор бұрын сақтаған үйінділерді автоматты түрде босатады, файлдарды жабады және т.б. операциялар. Бағдарламашы, егер бағынышты объектілер сақталған болса, кластың барлық даналары үшін деструкторларды шақыруға жауап береді.

Деструкторды іске асыруға арналған синтаксис:

Деструктор <сынып аты>. <Деструктор атауы> [(<параметрлер>)];

[<жарнама блогы>]

Баста

<орындалатын операторлар>

Соңы;

Тұқым қуалаушыларды іске асыру

Егер сіз деструкторлардың мұрагерлік механизмін қолданатын болсаңыз, онда сіз класс инстанцияларын жоюды жеңілдете аласыз, осылайша сіз осы сыныпқа қосылған өрістерді жою туралы ғана ойлайсыз. Мұраланған өрістерді тазарту бойынша барлық жұмыстар мұрагерлік деструкторлармен жасалуы мүмкін. Мұраланған деструкторды шақыру үшін мұрагерлік кілт сөзін қолдану керек.

Тұқым қуалаушылық деструкторды жариялау синтаксисі келесідей:

Деструктор <сынып аты>. <Деструктор атауы> [(<параметрлер>)};

[<жарнама блогы>]

Баста

<меншікті өрістерді жою>

Мұраға қалдырылған <деструктор атауы> [{<параметрлер>)];

Соңы;

Ескертулер:

• Деструктордың ішінде қарапайым идентификаторларға ғана емес, сонымен қатар анықталған кезде инкапсуляцияланған класс данасының өрістеріне де қол жетімді.

• Орындалатын деструктивті операторлар сыныптың данасын жою үшін қажет барлық тазалау операцияларын орындауы керек. Деструктивті код барлық ішкі объект даналарын жойып, сынып данасының өмір сүру кезеңінде сақталған үйіндіден босатуы керек. Алайда, тікелей қол жеткізу өрісін нөлге теңестірудің қажеті жоқ.

• Сізге сирек параметрлері бар деструкторларды жариялау қажет. Әдетте, деструктор тек класстың данасын жоюы керек, және оған қажет барлық ақпарат оған қол жетімді.

• Егер сыныпта жанама қатынау өрістері болмаса, онда деструкторды мұндай тапсырыс бойынша жасау қажет емес.

Конструкторларды шақыру

Конструкторды шақыру үшін класты дұрыс жариялап, анықтау керек. Класс декларациясы қол жетімді болуы керек, яғни. ол конструктор шақырылатын жерден болуы керек.

Егер теңшелетін сыныпта конструктор анықталмаса, онда әдепкі бойынша ұрпақ классынан қалған конструктор қолданылады. Кез келген жағдайда барлық объектілер TObject сыныбында анықталған Create конструкторына қол жеткізе алады.

Кластың анықтамасы сол сыныптың өкілдерін құра алатын белсенді құрылым жасайды. Класс анықтамасының көмегімен жасалынатын объектілер жаңадан құрылған объектілерге сілтемелерді сақтауға қабілетті.

Var <нысан атауы>: <сынып атауы>; // Меңзер айнымалысын жариялау

Баста

<объект атауы>: = <сынып аты>. <конструктор атауы> [(<параметрлер>)];

• Егер сіз конструкторды объект атынан шақырсаңыз, онда жаңа объект құрылмайды (жады бөлінбейді), бірақ конструктор кодында көрсетілген операторлар орындалады.

• Конструкторды класс көрсеткішінің түрінің айнымалысын қолдану арқылы да шақыруға болады.

[asd2]

Деструкторларды шақыру

Деструкторлар кластың көптеген басқа әдістерімен дәл осылай аталады - оның нақты данасы арқылы.

Конструкторды шақыруға арналған синтаксис келесідей:

<нысан атауы>. <жойғыш атауы> [(<параметрлер>)];

Ескертулер:

• Деструкторды шақырғаннан кейін объект қол жетімсіз болып қалады. Нөлді жойғаннан кейін оны бірден Nil-ге орнатқан жөн, содан кейін оның бар-жоғын тексеруге болады.

• Деструкторларды тікелей шақырмаңыз. TObject.Free-ден мұрагер болатын Free әдісін шақыру, Destroy's destructor шақырмас бұрын дананың нұсқауын Nil-мен салыстырады.

Delphi визуалды компоненттерінің конструкторлары мен деструкторларына қоңырау. Визуалды жобалау кезінде сіздің қосымшаңызға енетін кез-келген компонент формада жабылатын белгілі бір иерархияға кіреді (TFonn сыныбы). Сондықтан форманың барлық компоненттерінің конструкторлары мен деструкторларының шақыруы бағдарламалаушыға көрінбейтін форманы инициализациялау және жою кезінде автоматты түрде орындалады.

Пішіндердің өздері қосымша арқылы құрылады және жойылады - Application деп аталатын ғаламдық объект. \* .Dpr кеңейтімі бар жоба файлында форма конструкторына шақыруды жол ретінде көруге болады:

Application.CreateForm (TForml, Fonnl);

Объектілерді динамикалық құру. Пайдаланушы объект құра алады және бағдарламалық түрде:

Var Mem: TMemo;

Баста

Mem: = TMemo.Create (Өзі); // TMemo сыныбының данасын жасаңыз

Mem.Parent: = Өзі; // Form1 міндетті емес

Mem.Name:='TmpMem '; // Компонентке атау беру

FindComponent (‘TmpMem’). Тегін; // Компонентті алып тастаңыз

**Студенттердің білімін, іскерлік пен дағдысын қалыптастыру және бекіту, (пысықтау), сабақты қорытындылау:**

**Сабақты қорытындылау:** Сабаққа белсенді қатысып тапсырманы орындаған студенттердің білімін бағалау және баға қою

**Үйге тапсырма.**

**Бақылау сұрақтары**

1. Delphi ОБП тілі қандай программалау тіліне жатады?
2. Объектілі- бағдарланған программалаудың негізгі түсініктері қандай?
3. Delphi тілінің интеграцияланған ортасы қандай терезелерден құралады?
4. Объект инспекторының терезесі қандай астарлы беттерден тұрады, олардың қызметі.