Федеральное государственное автономное образовательное учреждение

высшего образования

«Казанский (Приволжский) федеральный университет»

Институт вычислительной математики и информационных технологий

**Курсовая работа**

**по дисциплине**

**«Базы данных»**

Выполнила студентка группы 09-612

*Айзатуллина А. Ф.*

Проверил: *Салимов Ф. И.*

Казань, 2018

**Описание базы данных**

**ТАБЛИЦА 2\_1. ПОДПИСЧИКИ**

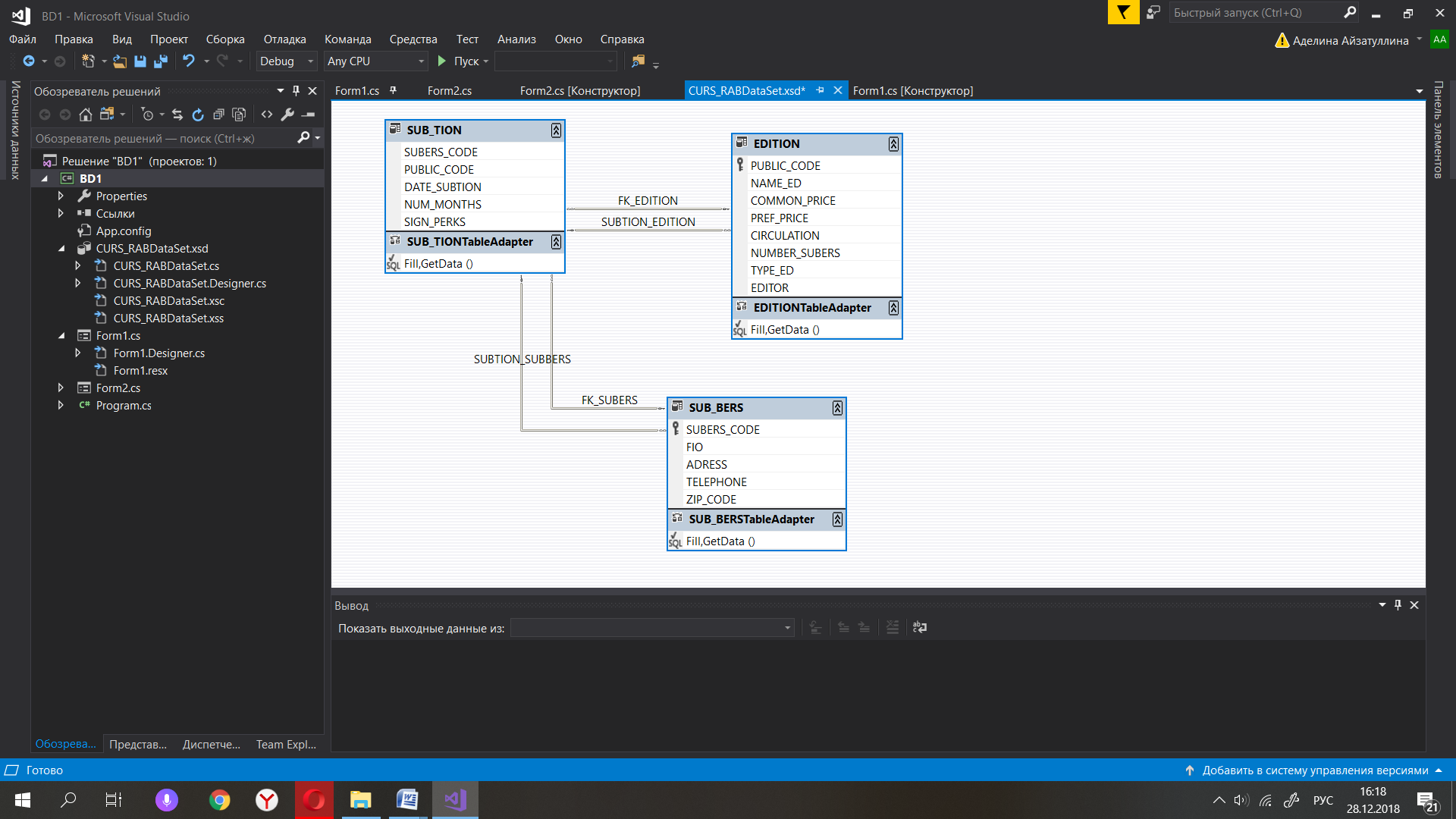
1. Код подписчика
2. ФИО
3. Место жительства (населенный пункт)
4. Номер телефона
5. Почтовый индекс

**ТАБЛИЦА 2\_2. ИЗДАHИЯ**

1. код издания
2. название
3. цена за месяц(обычная)
4. цена за месяц(льготная)
5. тираж
6. Число подписчиков (связать с таблицей Подписка)
7. тип издания (жуpнал, газета, книга...)
8. ФИО редактора

**ТАБЛИЦА 2.3. ПОДПИСКА**

1. код подписчика
2. код издания
3. дата подписки
4. на сколько месяцев
5. признак льготы



При создании базы данных предусмотреть:

Описание таблиц должно включать использование (хотя бы по одному разу): *NOT NULL, DEFAULT, PRIMARY KEY и CHECK* Использование доменов при определении таблиц является желательным.

Описание межтабличных связей (можно использовать средства каскадного удаления и редактирования, но хотя бы для одной связи эти средства должны быть запрограммированы при помощи триггеров)

Подготовить *SQL-script* для загрузки данных в таблицы (не менее 10–ти строк в каждой таблице).

При помощи триггеров установить связь между полем «Число подписчиков» и Таблицей «Подписка»

**Вариант 2\_01.**

1) Найти и вывести сведения о подписке на журналы в заданном году (параметр):

* ФИО подписчика
* почтовый индекс
* место жительства
* название журнала
* Дата подписки

2.1) Найти и вывести список ФИО подписчиков согласно условию: в заданном году (параметр) они подписались не на все газеты из каталога изданий.

2.2) Вывести газеты, которые в 2015 году подписали все подписчики мужского пола

3) Для каждого подписчика найти и вывести:

* ФИО
* количество газет и количество журналов, на которые он подписался в заданном году (год является параметром)

### 1. Файл создания базы данных в MS SQL Server Express

CREATE DATABASE CURS\_RAB

GO

USE CURS\_RAB

GO

CREATE TABLE SUB\_BERS( /\*ПОДПИСЧИКИ\*/

SUBERS\_CODE INT IDENTITY(1,1) NOT NULL CONSTRAINT PK\_SUBERS PRIMARY KEY, /\*Код подписчика\*/

FIO VARCHAR(100) NOT NULL, /\*ФИО\*/

ADRESS VARCHAR(150) NOT NULL, /\*Место жительства (населенный пункт)\*/

TELEPHONE VARCHAR(30) NOT NULL, /\*Номер телефона\*/

ZIP\_CODE VARCHAR (10) NOT NULL, /\*Почтовый индекс\*/

);

CREATE TABLE EDITION( /\*ИЗДАHИЯ\*/

PUBLIC\_CODE INT IDENTITY(1,1) NOT NULL CONSTRAINT PK\_EDITION PRIMARY KEY, /\*код издания\*/

NAME\_ED VARCHAR(100) NOT NULL, /\*название\*/

COMMON\_PRICE INT DEFAULT (0) CONSTRAINT CH\_COMMON CHECK ([COMMON\_PRICE]>=(0)), /\*цена за месяц(обычная)\*/

PREF\_PRICE INT DEFAULT (0) CONSTRAINT CH\_PREF CHECK ([PREF\_PRICE]>=(0)), /\*цена за месяц(льготная)\*/

CIRCULATION INT DEFAULT (1) CONSTRAINT CH\_CIRCULATION CHECK ([CIRCULATION]>=(1)), /\*тираж\*/

NUMBER\_SUBERS INT NULL DEFAULT (0), /\*число подписчиков (связать с таблицей Подписка)\*/

TYPE\_ED VARCHAR(10) NOT NULL CONSTRAINT CK\_TYPE\_ED CHECK (TYPE\_ED IN ('журнал','газета','книга')), /\*тип издания (жуpнал, газета, книга)\*/

EDITOR VARCHAR(100), /\*ФИО редактора\*/

) ON [PRIMARY];

CREATE TABLE SUB\_TION( /\*ПОДПИСКА\*/

SUBERS\_CODE INT NOT NULL CONSTRAINT FK\_SUBERS FOREIGN KEY REFERENCES SUB\_BERS(SUBERS\_CODE) ON DELETE CASCADE ON UPDATE CASCADE, /\*код подписчика\*/

PUBLIC\_CODE INT NOT NULL CONSTRAINT FK\_EDITION FOREIGN KEY REFERENCES EDITION(PUBLIC\_CODE) ON DELETE CASCADE ON UPDATE CASCADE, /\*код издания\*/

DATE\_SUBTION DATE, /\*дата подписки\*/

NUM\_MONTHS INT NOT NULL DEFAULT (1) CONSTRAINT CH\_MONTHS CHECK ([NUM\_MONTHS]>=(1)), /\*на сколько месяцев\*/

SIGN\_PERKS VARCHAR (30) NOT NULL CONSTRAINT CK\_SIGN\_PERKS CHECK (SIGN\_PERKS = 'есть льгота' OR SIGN\_PERKS = 'нет льготы'), /\*признак льготы\*/

);

INSERT INTO SUB\_BERS (FIO, ADRESS, TELEPHONE, ZIP\_CODE)

VALUES

('Милорадов Поликарп Савелиевич', 'г. Большое Мурашкино, ул. Барклая, дом 26, квартира 83', '8 (974) 858-42-88', '659350,'),

('Круглова Ариадна Ростиславовна ', 'г. Ардатов, ул. Бакунинская, дом 93, квартира 285', '8 (925) 825-53-80', '422559'),

('Дёшина Агния Андрияновна', 'г. Рогнедино, ул. Загородная, дом 33, квартира 181', '8 (933) 907-53-95', '416109'),

('Ермилов Макар Ираклиевич', 'г. Снежинск, ул. Башиловская, дом 95, квартира 126', '8 (975) 902-42-13', '215472'),

('Серпионов Богдан Иванович', 'г. Сочи, ул. Хрущёва, дом 90, квартира 284', '8 (928) 293-35-37', '652392'),

('Янышева Марфа Федоровна', 'г. Сочи, ул. Барышиха, дом 86, квартира 95', '8 (952) 211-83-55', '352217'),

('Берия Клавдий Евграфович', 'г. Чердынь, ул. Бахрушина, дом 64, квартира 171', '8 (970) 367-84-59', '659300'),

('Погребняк Измаил Эдуардович', 'г. Касторное, ул. Баррикадная, дом 68, квартира 26', '8 (902) 554-15-85', '141273'),

('Масмехова Дарья Ильевна ', 'г. Усть-Джегута, ул. Завокзальная 2-я, дом 46, квартира 267', '8 (965) 844-14-96', '422520'),

('Андрюшина Людмила Казимировна', 'г. Уйское, ул. Барболина, дом 1, квартира 141', '8 (940) 395-31-40', '624582'),

('Ревякин Феофан Никитевич', ' г. Шаранга, ул. Гайдара, дом 60, квартира 14', '8 (904) 920-56-82', '667008'),

('Ярушина Александра Леонидовна', 'г. Майкоп, ул. Бебеля, дом 68, квартира 81', '8 (928) 585-67-73', '182725'), /\*12\*/

('Доронин Михаил Алексеевич', 'г. Казань, ул. Барклая, дом 26, квартира 83', '8 (974) 858-42-88', '659350,'),

('Магина Анастасия Андреевна', 'г. Казань, ул. Бакунинская, дом 93, квартира 285', '8 (925) 825-53-80', '422559');

INSERT INTO SUB\_BERS (FIO, ADRESS, TELEPHONE, ZIP\_CODE)

VALUES

('Серегина Ксения Андреевна', 'г. Казань, ул. Бакунинская, дом 94, квартира 280', '8 (925) 825-53-80', '422559');

INSERT INTO EDITION (NAME\_ED, COMMON\_PRICE, PREF\_PRICE, CIRCULATION, TYPE\_ED, EDITOR)

VALUES

('1001 Совет женщинам', '150', '130', '5000', 'журнал', 'Фомов Анатолий Денисович'),

('Вязание для вас', '180', '150', '10000', 'журнал', 'Савельева Маргарита Иосифовна '),

('Великие российские предприниматели', '150', '100', '2000', 'книга', 'Чаурина Клеопатра Ярославовна'),

('Фантастика 20го века', '200', '170', '23000', 'книга', 'Русин Парамон Филиппович'),

('Собрание трактатов философов СССР', '165', '150', '85400', 'книга', 'Крылова Валентина Георгиевна '),

('Cosmopolitan', '200', '180', '1300000', 'журнал', 'Беляева Казимира Ильинична '),

('THE ART NEWSPAPER RUSSIA', '290', '250', '400000', 'газета', 'Белозёров Павел Викторович'); /\*7\*/

('777', '150', '125', '4000', 'газета', 'Серегина Ксения Иванович')

INSERT INTO SUB\_TION (SUBERS\_CODE, PUBLIC\_CODE, DATE\_SUBTION, NUM\_MONTHS, SIGN\_PERKS)

VALUES

('1', '7','01.04.2018', '8', 'есть льгота'),

('5', '7', '05.04.2018','7', 'нет льготы'),

('12', '5', '13.05.2018', '10', 'нет льготы'),

('12', '2', '25.06.2018', '6', 'нет льготы'),

('2', '2', '07.07.2018', '5', 'есть льгота'),

('5', '4', '10.07.2018', '6', 'нет льготы'),

('11', '1', '15.07.2018', '7', 'есть льгота'),

('3', '3', '30.07.2018', '9', 'есть льгота'),

('10', '6', '05.08.2018', '4', 'нет льготы'),

('4', '1', '14.08.2018', '3', 'есть льгота'),

('8', '1', '15.08.2018', '7', 'нет льготы'),

('9', '2', '17.08.2018', '6', 'нет льготы'),

('6', '1', '24.08.2018', '2', 'есть льгота'),

('7', '3', '19.09.2018', '4', 'есть льгота'),

('4','3', '25.10.2018', '1', 'есть льгота');

('13', '9','01.05.2015', '8', 'есть льгота')

('13', '8', '05.04.2015','7', 'нет льготы'),

('14', '8', '05.04.2015','7', 'нет льготы');

## 2. При помощи триггеров установить связь между полем «Число подписчиков» и Таблицей «Подписка»

CREATE TRIGGER AI\_SUB\_TION ON SUB\_TION

AFTER INSERT AS

BEGIN

DECLARE @COUNT INT;

SET @COUNT = @@ROWCOUNT; /\*проверка, что ни одна строка не была затронута операцией вставки\*/

IF @COUNT = 0 /\*count создали, потому что rowcount постоянно сбрасывается\*/

RETURN;

SET NOCOUNT ON;

UPDATE EDITION

SET NUMBER\_SUBERS = NUMBER\_SUBERS + (SELECT COUNT(PUBLIC\_CODE) /\*НУЖЕН ЛИ EDITION?!\*/

FROM INSERTED i WHERE i.PUBLIC\_CODE = EDITION.PUBLIC\_CODE) /\*при выполнении триггера всегда создаются Inserted и Deleted.\*/

WHERE PUBLIC\_CODE IN (SELECT PUBLIC\_CODE FROM INSERTED i1);

END

/\*если подписка закончилась, нужно удалить подписку из таблицы\*/

CREATE TRIGGER BD\_SUB\_TION ON SUB\_TION

AFTER DELETE AS

BEGIN

DECLARE @COUNT INT;

SET @COUNT = @@ROWCOUNT;

IF @COUNT = 0

RETURN;

SET NOCOUNT ON;

UPDATE EDITION

SET NUMBER\_SUBERS = NUMBER\_SUBERS - (SELECT COUNT(PUBLIC\_CODE)

FROM DELETED d WHERE d.PUBLIC\_CODE = EDITION.PUBLIC\_CODE)

WHERE PUBLIC\_CODE IN (SELECT PUBLIC\_CODE FROM DELETED d1);

END

CREATE TRIGGER AU\_SUB\_TION ON SUB\_TION

AFTER UPDATE AS

BEGIN

DECLARE @COUNT INT;

SET @COUNT = @@ROWCOUNT;

IF @COUNT = 0

RETURN;

SET NOCOUNT ON;

IF UPDATE (PUBLIC\_CODE)

BEGIN

SELECT \* FROM DELETED

SET @COUNT = @@ROWCOUNT;

IF @COUNT > 0

UPDATE EDITION

SET NUMBER\_SUBERS = NUMBER\_SUBERS - (SELECT COUNT(PUBLIC\_CODE)

FROM DELETED d WHERE d.PUBLIC\_CODE = EDITION.PUBLIC\_CODE

GROUP BY PUBLIC\_CODE)

WHERE PUBLIC\_CODE IN(SELECT PUBLIC\_CODE FROM DELETED d1)

SELECT \* FROM INSERTED

SET @COUNT = @@ROWCOUNT;

IF @COUNT > 0

UPDATE EDITION

SET NUMBER\_SUBERS = NUMBER\_SUBERS + (SELECT COUNT(PUBLIC\_CODE)

FROM INSERTED i WHERE i.PUBLIC\_CODE = EDITION.PUBLIC\_CODE

GROUP BY PUBLIC\_CODE)

WHERE PUBLIC\_CODE IN (SELECT PUBLIC\_CODE FROM INSERTED i1);

END

END;

## 3. View

USE CURS\_RAB

GO

use CinemaProkat

GO

CREATE VIEW Film AS SELECT\* FROM Films WHERE prize > 0;

GO

CREATE VIEW MY\_CATALOG (NAME\_ED, TYPE\_ED, EDITOR, COMMON\_PRICE, PREF\_PRICE)

AS

SELECT EDITION.NAME\_ED AS 'Название издания',

EDITION.TYPE\_ED AS 'Тип издания',

EDITION.EDITOR AS 'Автор',

EDITION.COMMON\_PRICE AS 'Обычная цена',

EDITION.PREF\_PRICE AS 'Льготная цена'

FROM EDITION;

CREATE VIEW LGOTNIKI\_1001SOVET (FIO, ADRESS, TELEPHONE, ZIP\_CODE)

AS

SELECT SUB\_BERS.FIO AS 'ФИО',

SUB\_BERS.ADRESS AS 'Адрес',

SUB\_BERS.TELEPHONE AS 'Телефон',

SUB\_BERS.ZIP\_CODE AS 'Индекс'

FROM EDITION, SUB\_BERS, SUB\_TION

WHERE SUB\_BERS.SUBERS\_CODE = SUB\_TION.SUBERS\_CODE

AND EDITION.PUBLIC\_CODE = SUB\_TION.PUBLIC\_CODE

AND EDITION.NAME\_ED = '1001 Совет женщинам'

AND SUB\_TION.SIGN\_PERKS = 'есть льгота';

## 4. Основная экранная форма

Form1 - экранная форма, на которой отображены все таблицы, при клике на определенную таблицу Label и Navigator переключаются на нее. Кнопки определены для таблицы Cinemas. Добавлена альтернативная экранная форма, которая открывается при клике

MenuStrip->Формы->альтернативная

Между таблицами Cinemas<->>Prokat <<->Films установила реляцию. При клике на строку таблицы Prokat в родительских таблицах отображаются только записи, связанные с этой строкой.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace KursPab

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_TION". При необходимости она может быть перемещена или удалена.

this.sUB\_TIONTableAdapter.Fill(this.cursRabDataSet.SUB\_TION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION". При необходимости она может быть перемещена или удалена.

this.eDITIONTableAdapter.Fill(this.cursRabDataSet.EDITION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_BERS". При необходимости она может быть перемещена или удалена.

this.sUB\_BERSTableAdapter.Fill(this.cursRabDataSet.SUB\_BERS);

this.Text = "Основная форма";

}

private void sUBBERSGridView\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

bindingNavigator1.BindingSource = sUBBERSBindingSource;

lblTable.Text = "Подписчики";

this.sUBBERSGridView.DataSource = sUBBERSBindingSource;

}

private void eDITIONGridView\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

bindingNavigator1.BindingSource = eDITIONBindingSource;

lblTable.Text = "Издания";

this.eDITIONGridView.DataSource = eDITIONBindingSource;

}

private void sUBTIONGridView\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

sUBTIONGridView.DataSource = sUBTIONBindingSource;

// this.eDITIONGridView.DataSource = sUBTION\_EDITIONBindingSource;

// this.sUBBERSGridView.DataSource = sUBTION\_SUBBERSBindingSource;//это обратные связи

lblTable.Text = "Подписки";

}

private void sUBTIONGridView\_CurrentCellChanged(object sender, EventArgs e)

{

if (sUBTIONGridView.CurrentCell != null)

{

int nRow = sUBTIONGridView.CurrentCell.RowIndex;

//первая строка

if (nRow == 0)

{

btnPrev.Enabled = false;

btnFirst.Enabled = false;

}

else

{

btnPrev.Enabled = true;

btnFirst.Enabled = true;

}

//последняя строка

if (nRow == sUBTIONGridView.RowCount - 1)

{

btnNext.Enabled = false;

btnLast.Enabled = false;

}

else

{

btnNext.Enabled = true;

btnLast.Enabled = true;

}

}

}

private void btnFirst\_Click(object sender, EventArgs e)

{

lblTable.Text = "Подписки";

int nCol = sUBTIONGridView.CurrentCell.ColumnIndex;

sUBTIONGridView.CurrentCell = sUBTIONGridView[nCol, 0];

btnPrev.Enabled = btnFirst.Enabled = false;

btnLast.Enabled = btnNext.Enabled = true;

}

private void btnLast\_Click(object sender, EventArgs e)

{

lblTable.Text = "Подписки";

int nCol = sUBTIONGridView.CurrentCell.ColumnIndex;

sUBTIONGridView.CurrentCell = sUBTIONGridView[nCol, sUBTIONGridView.RowCount - 2];

btnPrev.Enabled = btnFirst.Enabled = true;

btnLast.Enabled = btnNext.Enabled = false;

}

private void btnPrev\_Click(object sender, EventArgs e)

{

lblTable.Text = "Подписки";

int nRow = sUBTIONGridView.CurrentCell.RowIndex;

int nCol = sUBTIONGridView.CurrentCell.ColumnIndex;

if (nRow > 0)

sUBTIONGridView.CurrentCell = sUBTIONGridView[nCol, --nRow];

if (nRow == 0)

btnPrev.Enabled = btnFirst.Enabled = false;

if (nRow != sUBTIONGridView.RowCount - 2)

btnLast.Enabled = btnNext.Enabled = true;

}

private void btnNext\_Click(object sender, EventArgs e)

{

lblTable.Text = "Подписки";

int nRow = sUBTIONGridView.CurrentCell.RowIndex;

int nCol = sUBTIONGridView.CurrentCell.ColumnIndex;

if (nRow >= 0)

sUBTIONGridView.CurrentCell = sUBTIONGridView[nCol, ++nRow];

if (nRow != 0)

btnPrev.Enabled = btnFirst.Enabled = true;

if (nRow == sUBTIONGridView.RowCount - 2)

btnLast.Enabled = btnNext.Enabled = false;

}

private void btnSave\_Click(object sender, EventArgs e)

{

this.sUB\_TIONTableAdapter.Update(this.cursRabDataSet.SUB\_TION);

this.sUB\_TIONTableAdapter.Fill(this.cursRabDataSet.SUB\_TION);

this.eDITIONTableAdapter.Update(this.cursRabDataSet.EDITION);

this.eDITIONTableAdapter.Fill(this.cursRabDataSet.EDITION);

this.sUBTIONGridView\_CurrentCellChanged(sUBTIONGridView, e);

MessageBox.Show("Изменения сохранены");

}

private void btnRefres\_Click(object sender, EventArgs e)

{

this.sUB\_TIONTableAdapter.Fill(cursRabDataSet.SUB\_TION);

this.sUBTIONGridView\_CurrentCellChanged(sUBTIONGridView, e);

}

private void btnDelete\_Click(object sender, EventArgs e)

{

foreach (DataGridViewRow row in sUBTIONGridView.SelectedRows)

{

sUBTIONGridView.Rows.Remove(row);

}

}

private void btnAdd\_Click(object sender, EventArgs e)

{

btnPrev.Enabled = btnFirst.Enabled = true;

btnLast.Enabled = btnNext.Enabled = false;

sUBTIONGridView.CurrentCell = sUBTIONGridView[0, sUBTIONGridView.RowCount - 1];

}

private void bindingNavigator1\_RefreshItems(object sender, EventArgs e){ }

private void альтернативнаяToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Form2 f2 = new Form2();

f2.Show();

}

private void tableAdapterToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form3 f3 = new Form3();

f3.Show();

}

private void двойнойЦиклПервичныйКлючDataRelationToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form4 f4 = new Form4();

f4.Show();

}

private void работаСБДToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form5 f5 = new Form5();

f5.Show();

}

private void tableAdapterToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Form6 f6 = new Form6();

f6.Show();

}

private void тройнойЦиклToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form7 f7 = new Form7();

f7.Show();

}

private void tableAdapterToolStripMenuItem2\_Click(object sender, EventArgs e)

{

Form8 f8 = new Form8();

f8.Show();

}

private void подзапросыToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form9 f9 = new Form9();

f9.Show();

}

private void tableAdapterToolStripMenuItem3\_Click(object sender, EventArgs e)

{

Form10 f10 = new Form10();

f10.Show();

}

private void подзапросыToolStripMenuItem1\_Click(object sender, EventArgs e)

{

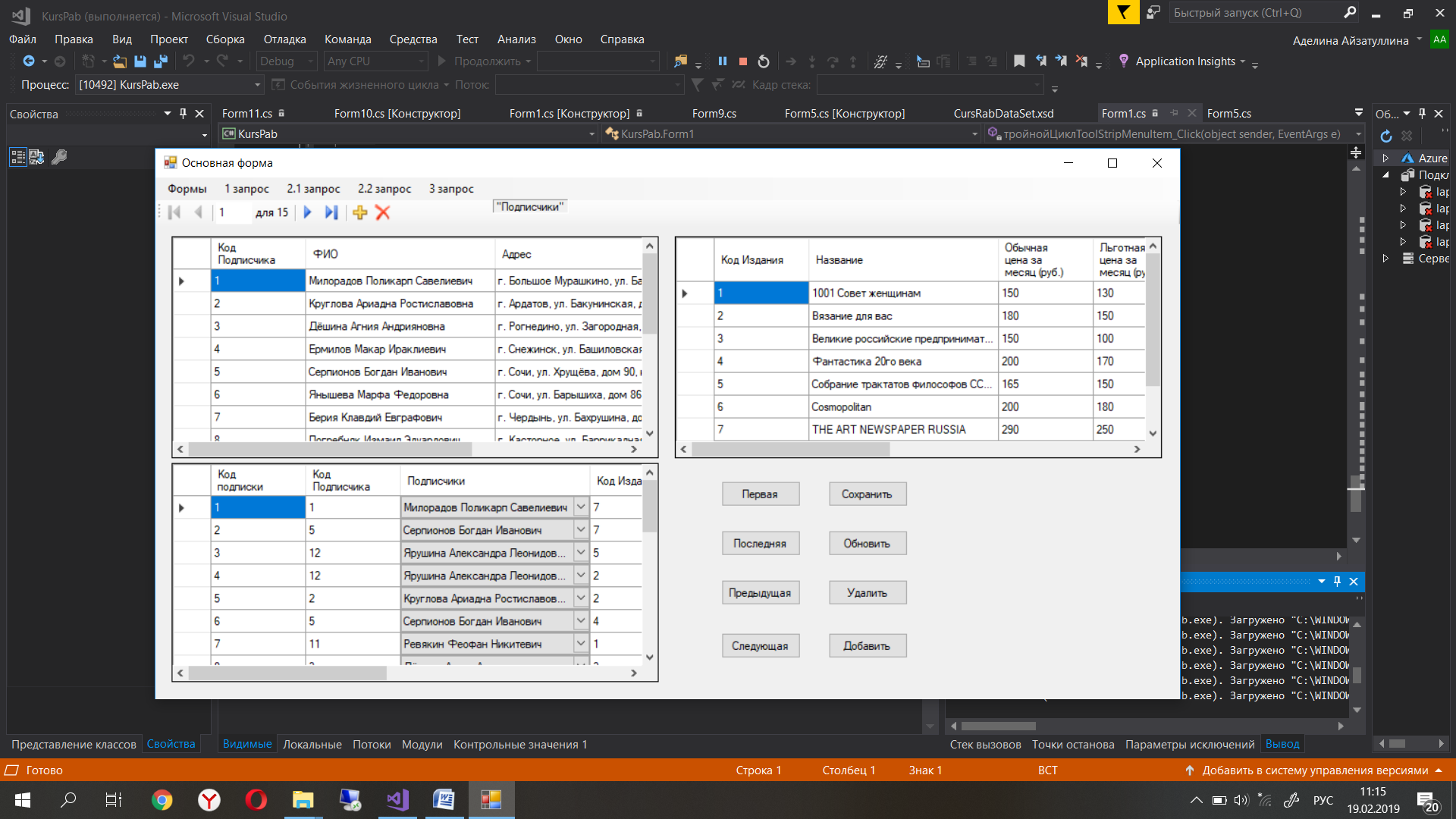
Form11 f11 = new Form11();

f11.Show();

}

}

}



## 4.2 Альтернативная экранная форма

Из альтернативной экранной формы пользователь может перейти к запросам через DataGridView один, но при клике на различные группы с именами таблиц, он меняется на соответствующую, как и label.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD1

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "cURS\_RABDataSet.SUB\_TION". При необходимости она может быть перемещена или удалена.

this.sUB\_TIONTableAdapter.Fill(this.cURS\_RABDataSet.SUB\_TION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cURS\_RABDataSet.SUB\_BERS". При необходимости она может быть перемещена или удалена.

this.sUB\_BERSTableAdapter.Fill(this.cURS\_RABDataSet.SUB\_BERS);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cURS\_RABDataSet.EDITION". При необходимости она может быть перемещена или удалена.

this.eDITIONTableAdapter.Fill(this.cURS\_RABDataSet.EDITION);

dataGridView1.AutoGenerateColumns = true;

}

private void btnEDITION\_Click(object sender, EventArgs e)

{

if (LabelName.Text != "Товары")

{

dataGridView1.Columns.Clear();

dataGridView1.DataSource = eDITIONBindingSource;

bindingNavigator1.BindingSource = eDITIONBindingSource;

LabelName.Text = "Товары";

}

}

private void btnSUB\_BERS\_Click(object sender, EventArgs e)

{

if (LabelName.Text != "Подписчики")

{

dataGridView1.Columns.Clear();

dataGridView1.DataSource = sUBBERSBindingSource;

bindingNavigator1.BindingSource = sUBBERSBindingSource;

LabelName.Text = "Подписчики";

}

}

private void btnSUB\_TION\_Click(object sender, EventArgs e)

{

if (LabelName.Text != "Подписки")

{

dataGridView1.Columns.Clear();

dataGridView1.DataSource = sUBTIONBindingSource;

bindingNavigator1.BindingSource = sUBTIONBindingSource;

LabelName.Text = "Подписки";

}

}

private void btnUpdate\_Click(object sender, EventArgs e)

{

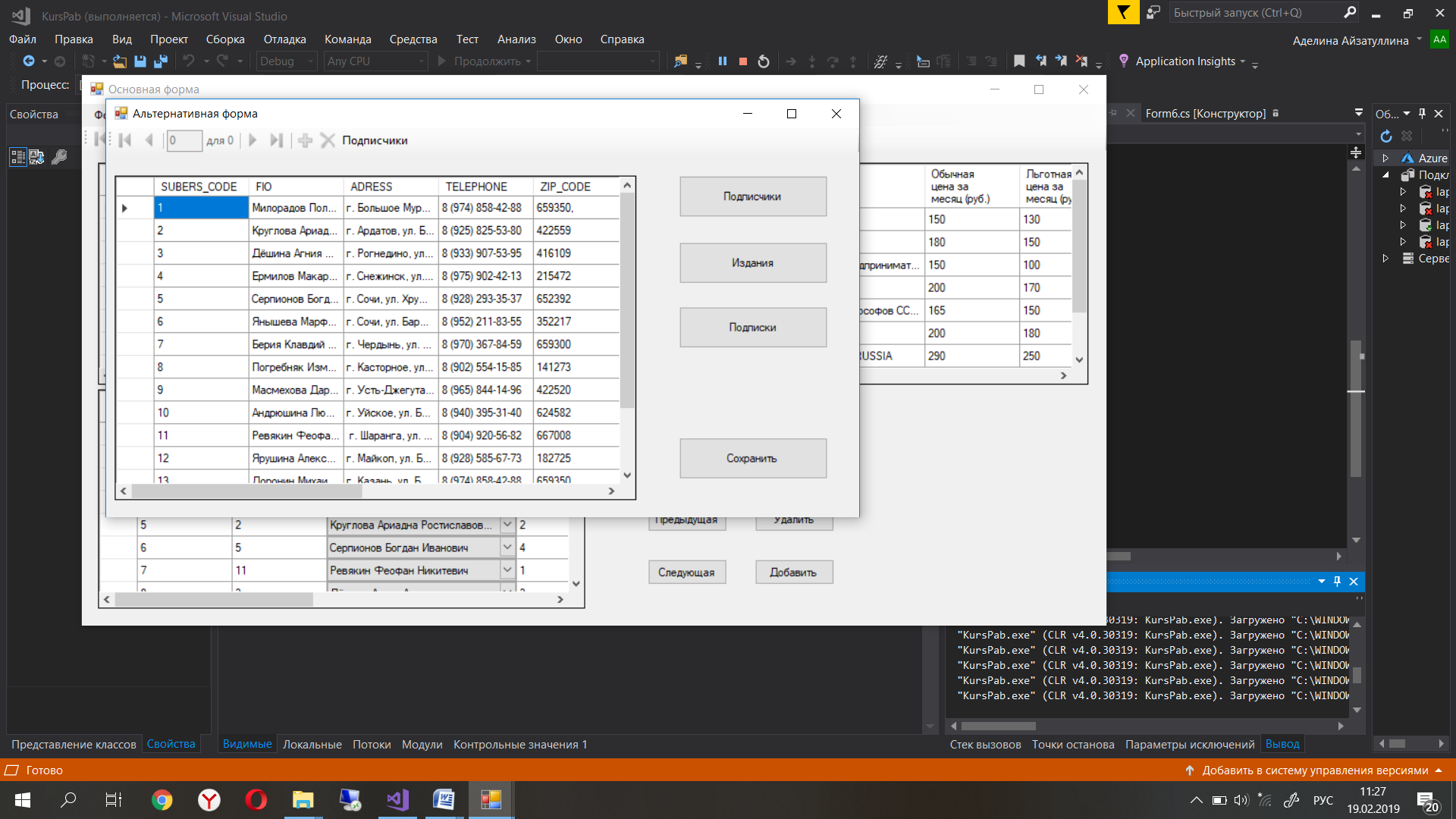
// tableAdapterManager.UpdateAll(cURS\_RABDataSet);

MessageBox.Show("Изменения сохранены");

}

}

}



## 5. Запросы

1) Найти и вывести сведения о подписке на журналы в заданном году (параметр):

* ФИО подписчика
* почтовый индекс
* место жительства
* название журнала
* Дата подписки

SELECT DISTINCT SUB\_BERS.FIO, SUB\_BERS.ZIP\_CODE, SUB\_BERS.ADRESS, EDITION.NAME\_ED, SUB\_TION.DATE\_SUBTION

FROM SUB\_BERS INNER JOIN

SUB\_TION ON SUB\_BERS.SUBERS\_CODE = SUB\_TION.SUBERS\_CODE INNER JOIN

EDITION ON SUB\_TION.PUBLIC\_CODE = EDITION.PUBLIC\_CODE

WHERE (EDITION.TYPE\_ED = 'журнал')

AND (YEAR(SUB\_TION.DATE\_SUBTION) = @YEAR)

ORDER BY SUB\_BERS.FIO



Form3 отображает выполнение первого запроса через SQL. В TextYEAR мы вводим параметр запроса – название нужного нам кинотеатра. Нажав на кнопку “Выполнить”, мы увидим нужный нам список фильмов.

using System;

…

namespace KursPab

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void Form3\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_TION". При необходимости она может быть перемещена или удалена.

this.sUB\_TIONTableAdapter.Fill(this.cursRabDataSet.SUB\_TION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_BERS". При необходимости она может быть перемещена или удалена.

this.sUB\_BERSTableAdapter.Fill(this.cursRabDataSet.SUB\_BERS);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION". При необходимости она может быть перемещена или удалена.

this.eDITIONTableAdapter.Fill(this.cursRabDataSet.EDITION);

this.Text = "TableAdapter";

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void btnFill1\_Click(object sender, EventArgs e)

{

int year = Convert.ToInt32(textYEAR.Text);

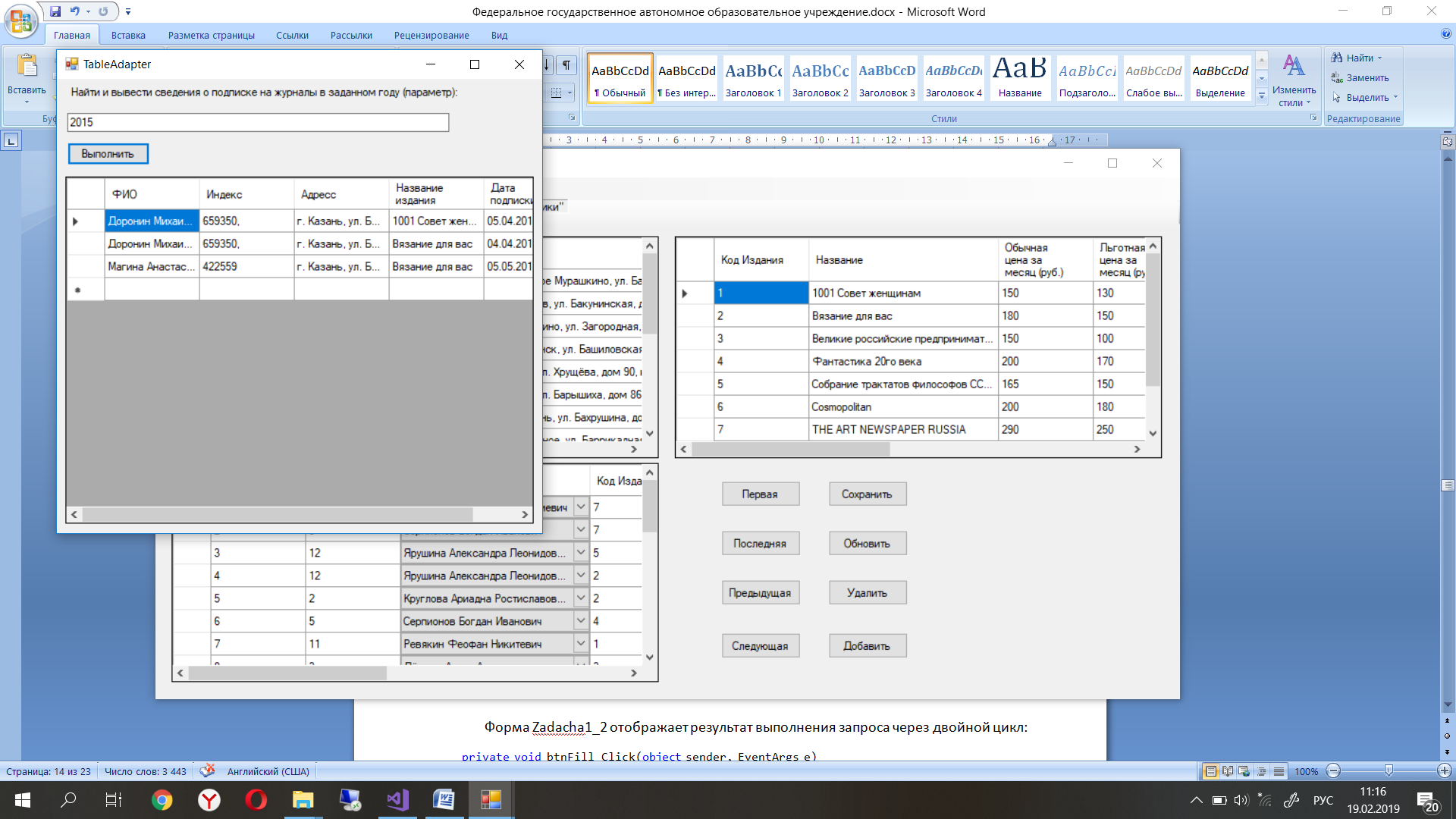
this.dataTable1TableAdapter.Fill(cursRabDataSet.DataTable1, year);

}

private void textYEAR\_TextChanged(object sender, EventArgs e) { }

}

}



Форма Zadacha1\_2 отображает результат выполнения запроса через двойной цикл:

private void btnFill\_Click(object sender, EventArgs e)

{

cURS\_RABDataSet.DataTable1.Clear();

int year = Convert.ToInt32(textYEAR.Text);

foreach (CURS\_RABDataSet.SUB\_TIONRow sRow in cURS\_RABDataSet.SUB\_TION.Rows)

{

if ((sRow.DATE\_SUBTION >= Convert.ToDateTime("01.01." + year)) && (sRow.DATE\_SUBTION <= Convert.ToDateTime("31.12." + year)))

{

foreach (CURS\_RABDataSet.EDITIONRow eRow in cURS\_RABDataSet.EDITION.Rows)

{

if (eRow.TYPE\_ED == "журнал")

{

if (eRow.PUBLIC\_CODE == sRow.PUBLIC\_CODE)

{

foreach (CURS\_RABDataSet.SUB\_BERSRow rRow in cURS\_RABDataSet.SUB\_BERS.Rows)

{

if (rRow.SUBERS\_CODE == sRow.SUBERS\_CODE)

{

CURS\_RABDataSet.DataTable1Row zRow = cURS\_RABDataSet.DataTable1.NewDataTable1Row();

zRow.FIO = rRow.FIO;

zRow.ZIP\_CODE = rRow.ZIP\_CODE;

zRow.ADRESS = rRow.ADRESS;

zRow.NAME\_ED = eRow.NAME\_ED;

zRow.DATE\_SUBTION = sRow.DATE\_SUBTION;

cURS\_RABDataSet.DataTable1.AddDataTable1Row(zRow);

}

}

}

}

}

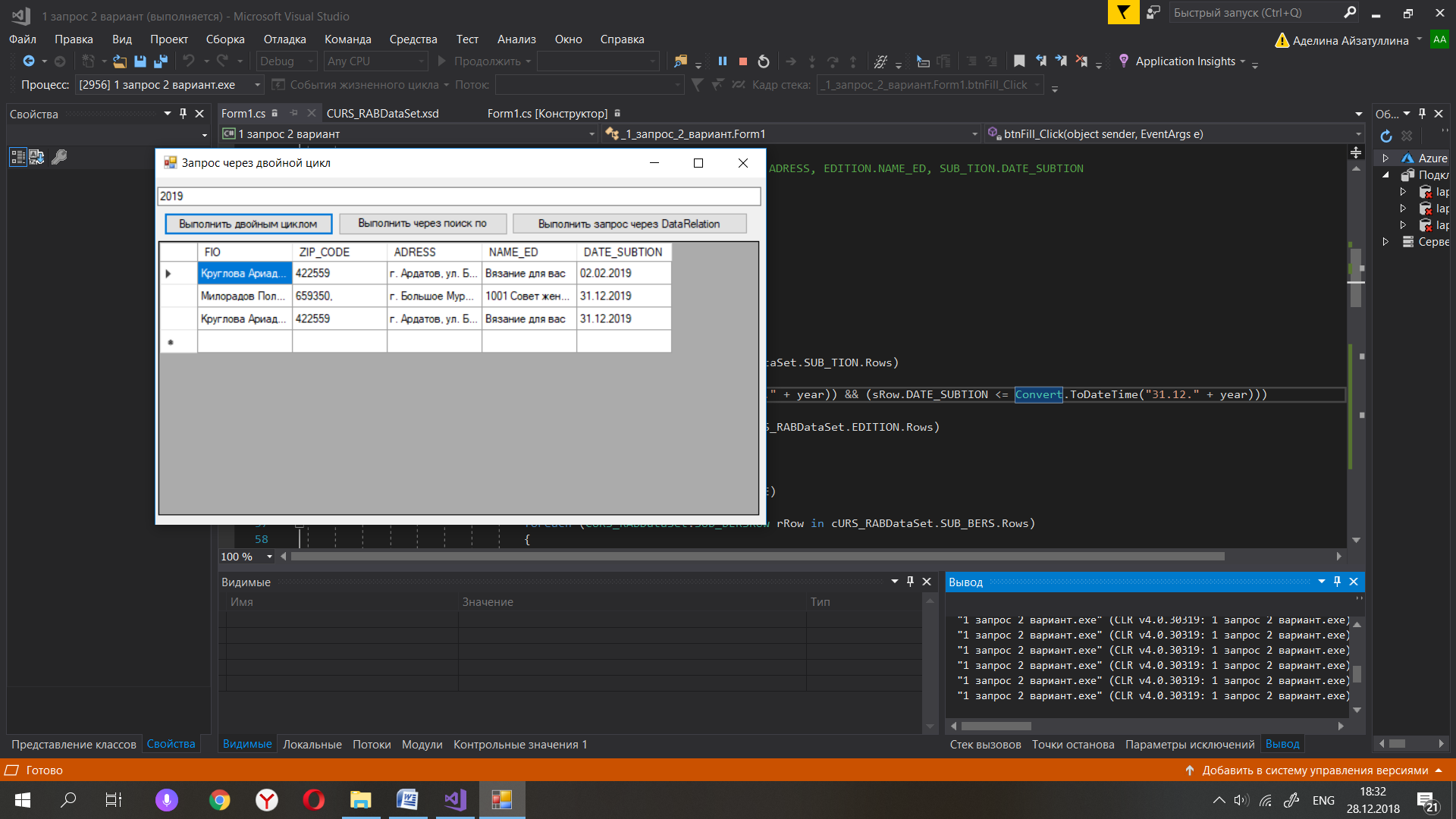
}

}

dataGridView1.Refresh();

this.Text = "Запрос через двойной цикл";

}



Форма Zadacha1\_3 отображает результат выполнения запроса через поиск по первичному ключу:

private void btnFill2\_Click(object sender, EventArgs e)

{

cURS\_RABDataSet.DataTable1.Clear();

int year = Convert.ToInt32(textYEAR.Text);

foreach (CURS\_RABDataSet.SUB\_TIONRow sRow in cURS\_RABDataSet.SUB\_TION.Rows)

{

if ((sRow.DATE\_SUBTION >= Convert.ToDateTime("01.01." + year)) && (sRow.DATE\_SUBTION <= Convert.ToDateTime("31.12." + year)))

{

CURS\_RABDataSet.EDITIONRow eRow = cURS\_RABDataSet.EDITION.Rows.Find(sRow.PUBLIC\_CODE) as CURS\_RABDataSet.EDITIONRow;

CURS\_RABDataSet.SUB\_BERSRow rRow = cURS\_RABDataSet.SUB\_BERS.Rows.Find(sRow.SUBERS\_CODE) as CURS\_RABDataSet.SUB\_BERSRow;

if (eRow.TYPE\_ED == "журнал")

{

CURS\_RABDataSet.DataTable1Row zRow = cURS\_RABDataSet.DataTable1.NewDataTable1Row();

zRow.FIO = rRow.FIO;

zRow.ZIP\_CODE = rRow.ZIP\_CODE;

zRow.ADRESS = rRow.ADRESS;

zRow.NAME\_ED = eRow.NAME\_ED;

zRow.DATE\_SUBTION = sRow.DATE\_SUBTION;

cURS\_RABDataSet.DataTable1.AddDataTable1Row(zRow);

}

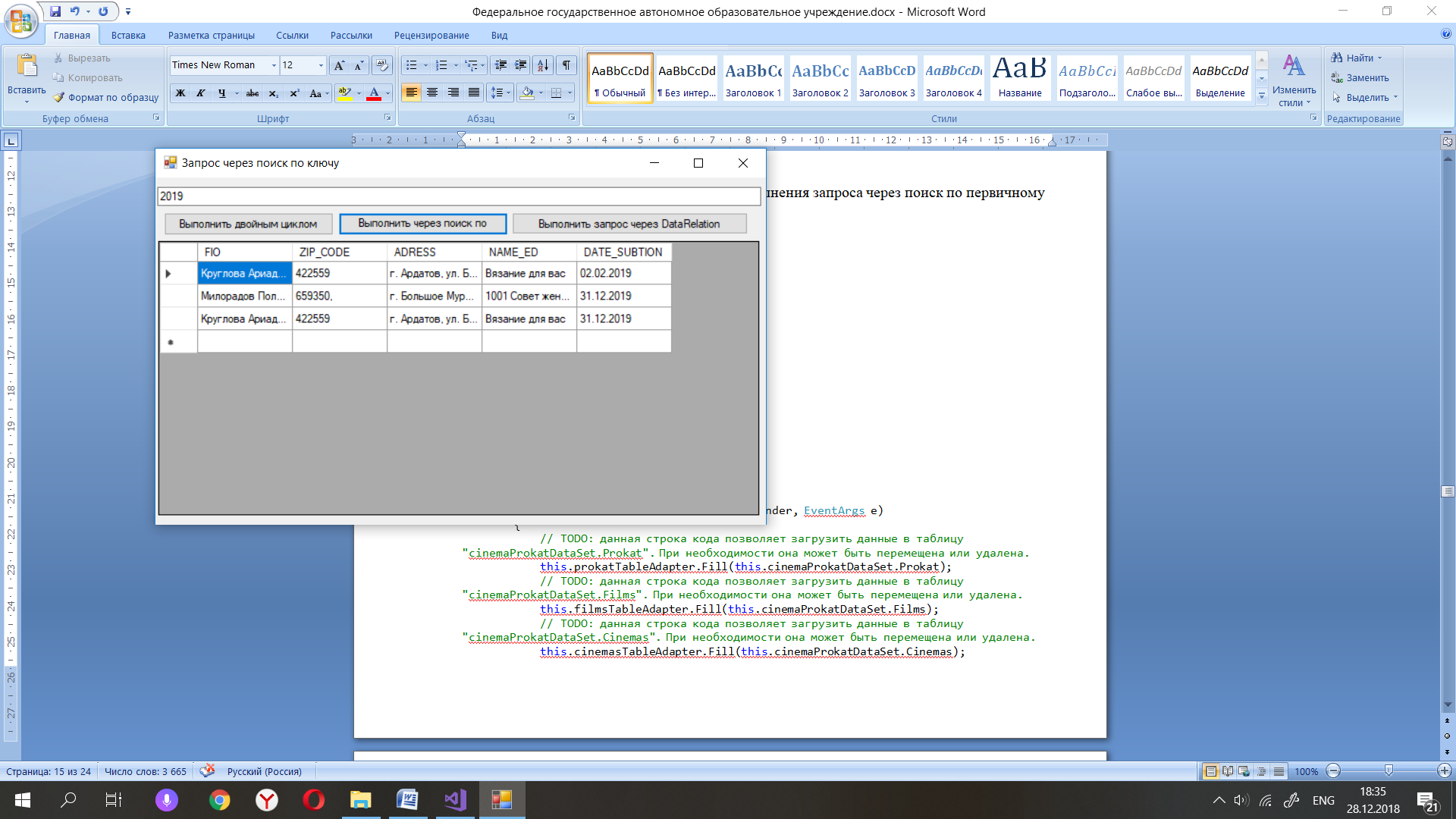
}

}

dataGridView1.Refresh();

this.Text = "Запрос через поиск по ключу";

}



Форма Zadacha1\_4 отображает результат выполнения запроса через DataRelation:

private void button1\_Click(object sender, EventArgs e)

{

cURS\_RABDataSet.DataTable1.Clear();

int year = Convert.ToInt32(textYEAR.Text);

foreach (CURS\_RABDataSet.SUB\_TIONRow sRow in cURS\_RABDataSet.SUB\_TION.Rows)

{

if ((sRow.DATE\_SUBTION >= Convert.ToDateTime("01.01." + year)) && (sRow.DATE\_SUBTION <= Convert.ToDateTime("31.12." + year)))

{

CURS\_RABDataSet.EDITIONRow eRow = sRow.GetParentRow("FK\_EDITION") as CURS\_RABDataSet.EDITIONRow;

CURS\_RABDataSet.SUB\_BERSRow rRow = sRow.GetParentRow("FK\_SUBERS") as CURS\_RABDataSet.SUB\_BERSRow;

if (eRow.TYPE\_ED == "журнал")

{

CURS\_RABDataSet.DataTable1Row zRow = cURS\_RABDataSet.DataTable1.NewDataTable1Row();

zRow.FIO = rRow.FIO;

zRow.ZIP\_CODE = rRow.ZIP\_CODE;

zRow.ADRESS = rRow.ADRESS;

zRow.NAME\_ED = eRow.NAME\_ED;

zRow.DATE\_SUBTION = sRow.DATE\_SUBTION;

cURS\_RABDataSet.DataTable1.AddDataTable1Row(zRow);

}

}

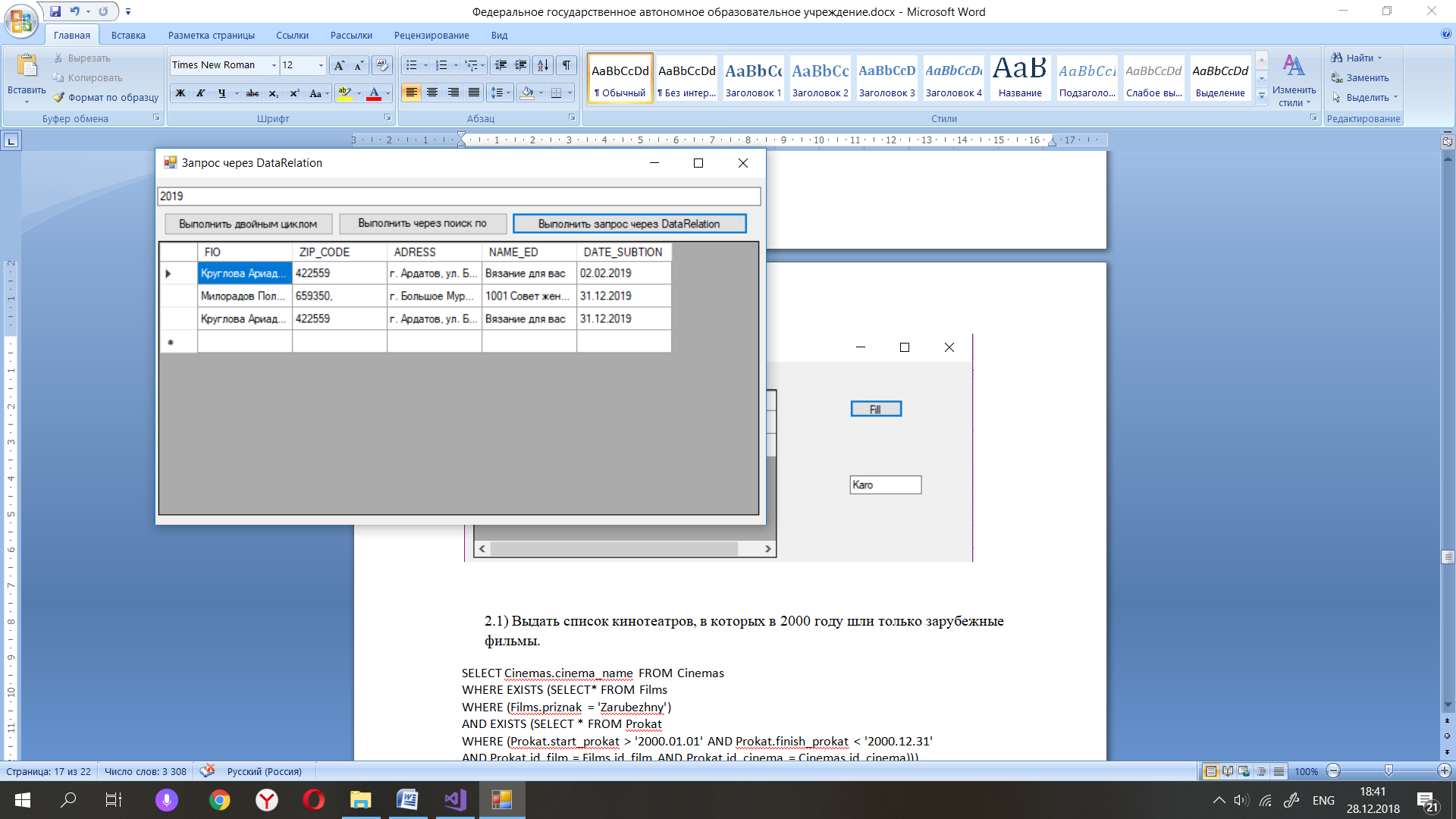
}

dataGridView1.Refresh();

this.Text = "Запрос через DataRelation";

}

}



Форма Zadacha1\_5 отображает результат работы программы через обращение непосредственно к базе данных и запросе на языке SQL:

using System;

…

namespace KursPab

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

private void Form5\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_TION". При необходимости она может быть перемещена или удалена.

this.sUB\_TIONTableAdapter.Fill(this.cursRabDataSet.SUB\_TION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_BERS". При необходимости она может быть перемещена или удалена.

this.sUB\_BERSTableAdapter.Fill(this.cursRabDataSet.SUB\_BERS);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION". При необходимости она может быть перемещена или удалена.

this.eDITIONTableAdapter.Fill(this.cursRabDataSet.EDITION);

this.Text = "Совместно с БД";

}

private void button1\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(Properties.Settings.Default.CursRabConnectionString);

con.Open();

int year = Convert.ToInt32(textYEAR.Text);

String command = "SELECT DISTINCT SUB\_BERS.FIO, SUB\_BERS.ZIP\_CODE, SUB\_BERS.ADRESS, EDITION.NAME\_ED, SUB\_TION.DATE\_SUBTION FROM SUB\_BERS, EDITION, SUB\_TION WHERE EDITION.PUBLIC\_CODE = SUB\_TION.PUBLIC\_CODE AND SUB\_BERS.SUBERS\_CODE = SUB\_TION.SUBERS\_CODE AND EDITION.TYPE\_ED = 'журнал'" +

"AND YEAR(SUB\_TION.DATE\_SUBTION) = " + year + ';';

SqlCommand cmd = new SqlCommand(command, con);

SqlDataReader rdr = cmd.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(rdr);

con.Close();

BindingSource bs = new BindingSource();

bs.DataSource = dt;

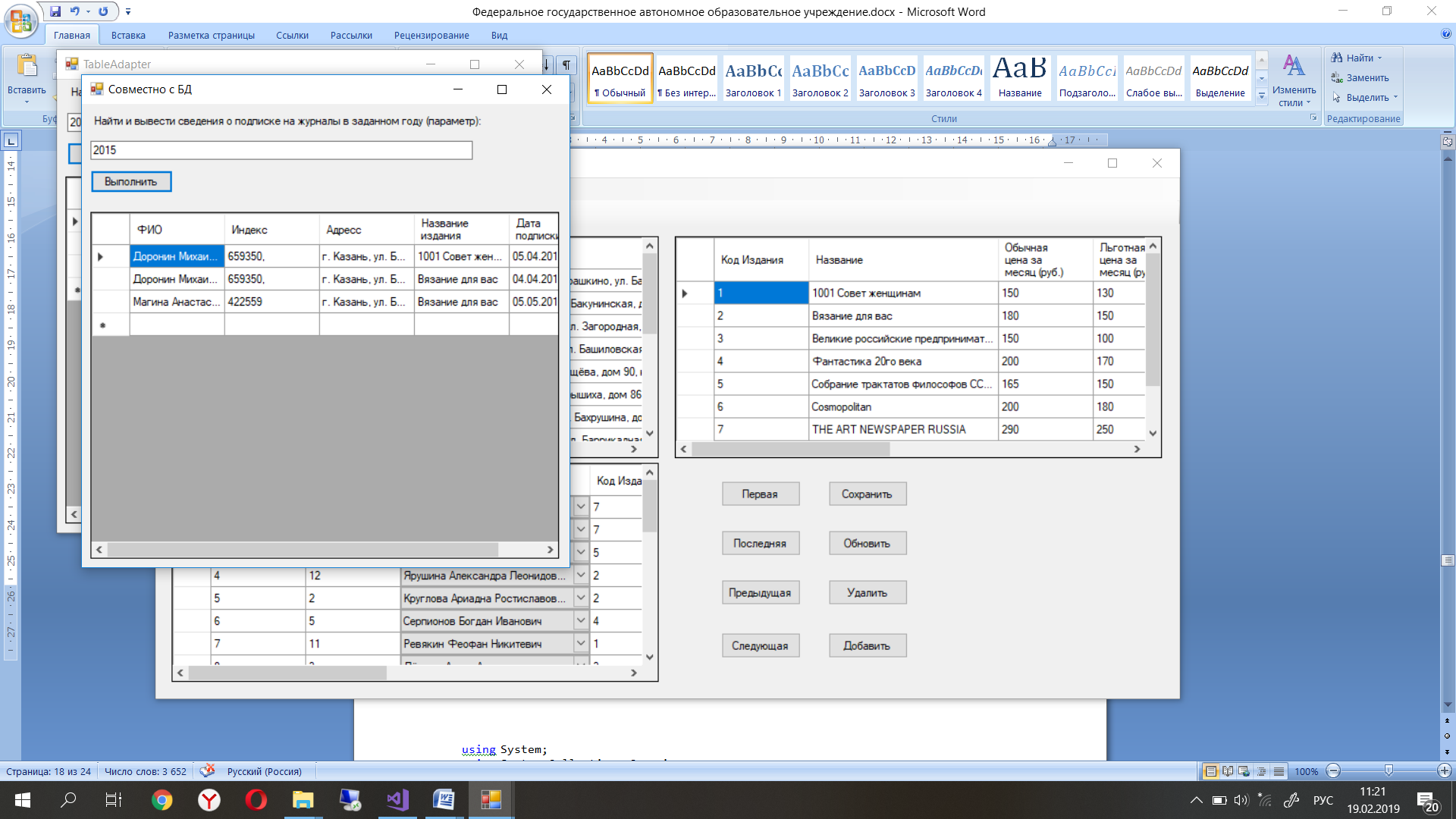
dataGridView1.DataSource = bs;

dataGridView1.Refresh();

}

}

}



2.1) Найти и вывести список ФИО подписчиков согласно условию:

в заданном году (параметр) они подписались не на все газеты из каталога изданий.

SELECT FIO FROM SUB\_BERS

WHERE EXISTS(SELECT \* FROM EDITION

WHERE (TYPE\_ED = 'газета')

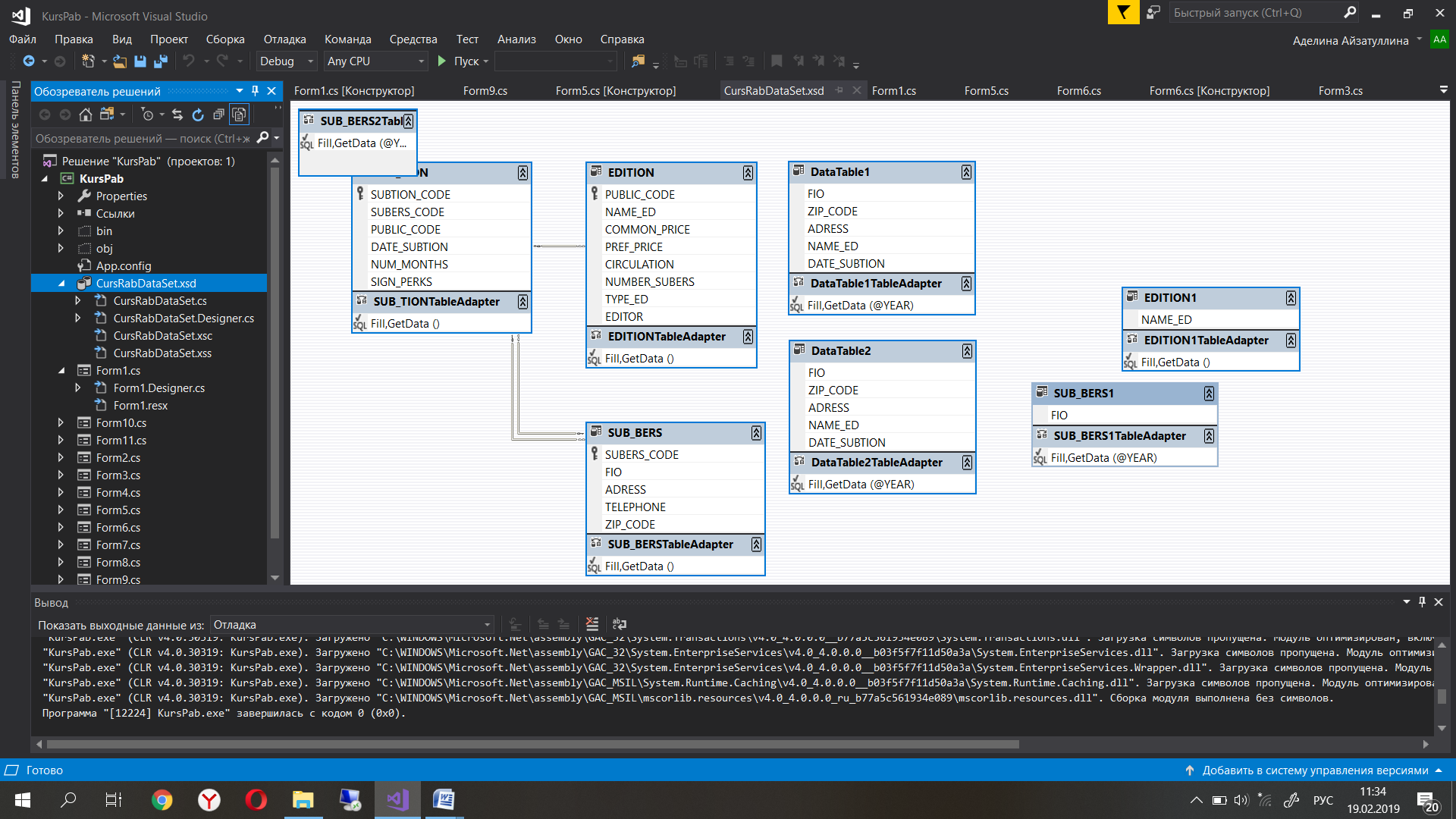
AND NOT EXISTS (SELECT \* FROM SUB\_TION

WHERE (EDITION.PUBLIC\_CODE = SUB\_TION.PUBLIC\_CODE)

AND (SUB\_BERS.SUBERS\_CODE = SUB\_TION.SUBERS\_CODE)

AND (YEAR(SUB\_TION.DATE\_SUBTION) = @YEAR))

)



Выполнение 2.1 запроса через SQL.

using System;

…

namespace KursPab

{

public partial class Form6 : Form

{

public Form6()

{

InitializeComponent();

}

private void Form6\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_TION". При необходимости она может быть перемещена или удалена.

this.sUB\_TIONTableAdapter.Fill(this.cursRabDataSet.SUB\_TION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_BERS". При необходимости она может быть перемещена или удалена.

this.sUB\_BERSTableAdapter.Fill(this.cursRabDataSet.SUB\_BERS);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION". При необходимости она может быть перемещена или удалена.

this.eDITIONTableAdapter.Fill(this.cursRabDataSet.EDITION);

dataGridView1.AutoGenerateColumns = true;

this.sUB\_BERS1TableAdapter.Fill(cursRabDataSet.SUB\_BERS1, YEAR: 0);

this.Text = "Запрос на основании объекта TableAdapter";

}

private void button1\_Click(object sender, EventArgs e)

{

int year = Convert.ToInt32(textYEAR.Text);

this.sUB\_BERS1TableAdapter.Fill(cursRabDataSet.SUB\_BERS1, year);

}

}

}

Выполнение при помощи вложенных циклов:

using System;

…

namespace KursPab

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

private void Form7\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_TION". При необходимости она может быть перемещена или удалена.

this.sUB\_TIONTableAdapter.Fill(this.cursRabDataSet.SUB\_TION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_BERS". При необходимости она может быть перемещена или удалена.

this.sUB\_BERSTableAdapter.Fill(this.cursRabDataSet.SUB\_BERS);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION". При необходимости она может быть перемещена или удалена.

this.eDITIONTableAdapter.Fill(this.cursRabDataSet.EDITION);

this.Text = "Запрос через тройной цикл";

}

private void button1\_Click(object sender, EventArgs e)

{

cursRabDataSet.SUB\_BERS1.Clear();

int year = Convert.ToInt32(textYEAR.Text);

bool aE, eS;

int max = 0;

int col = 0;

foreach (CursRabDataSet.EDITIONRow eRow in cursRabDataSet.EDITION.Rows)

{

if (eRow.TYPE\_ED == "газета") max++;

}

foreach (CursRabDataSet.SUB\_BERSRow rRow in cursRabDataSet.SUB\_BERS.Rows)

{

aE = true; //не любой журнал

foreach (CursRabDataSet.EDITIONRow eRow in cursRabDataSet.EDITION.Rows)

{

aE = (eRow.TYPE\_ED == "газета");

if (aE)

{

eS = false;

foreach (CursRabDataSet.SUB\_TIONRow sRow in cursRabDataSet.SUB\_TION.Rows)

{

eS = ((sRow.DATE\_SUBTION >= Convert.ToDateTime("01.01." + year)) && (sRow.DATE\_SUBTION <= Convert.ToDateTime("31.12." + year))

&& (eRow.PUBLIC\_CODE == sRow.PUBLIC\_CODE) && (rRow.SUBERS\_CODE == sRow.SUBERS\_CODE));

if (eS) break;

}

aE = eS;

}

if (aE) col++;

}

if (col != 0 && col != max)

// if(!aE||aE&&col!=max)

{

CursRabDataSet.SUB\_BERS1Row zRow = cursRabDataSet.SUB\_BERS1.NewSUB\_BERS1Row();

zRow.FIO = rRow.FIO;

cursRabDataSet.SUB\_BERS1.Rows.Add(zRow);

col = 0;

}

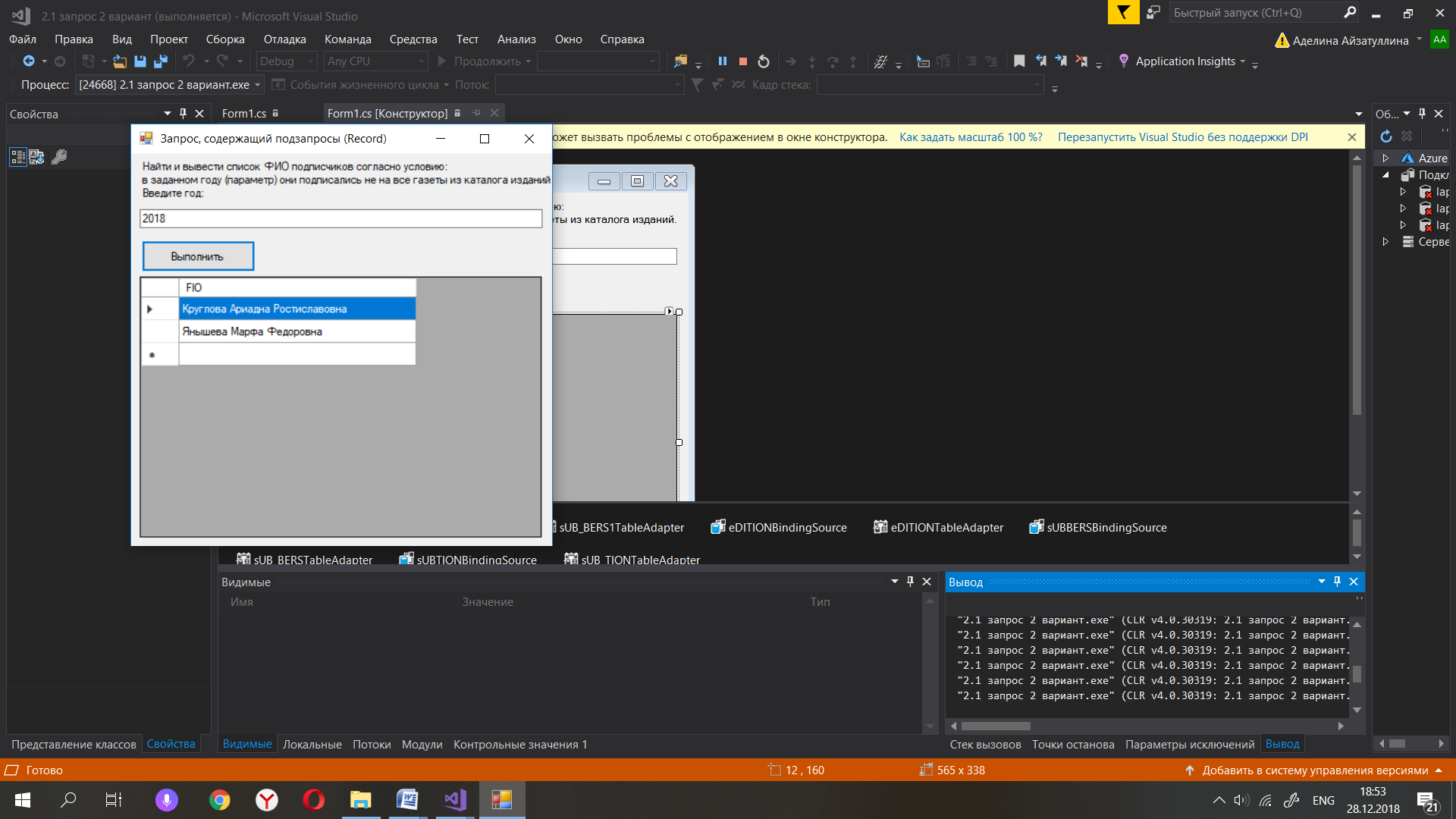
}

dataGridView1.Refresh();

}

}

}



2.2) Вывести газеты, которые в 2015 году подписали все подписчики из Казани

SELECT EDITION.NAME\_ED FROM EDITION

WHERE (TYPE\_ED = 'газета')

AND NOT EXISTS (SELECT \* FROM SUB\_BERS

WHERE(SUBSTRING(ADRESS, 4, 6) = 'Казань')

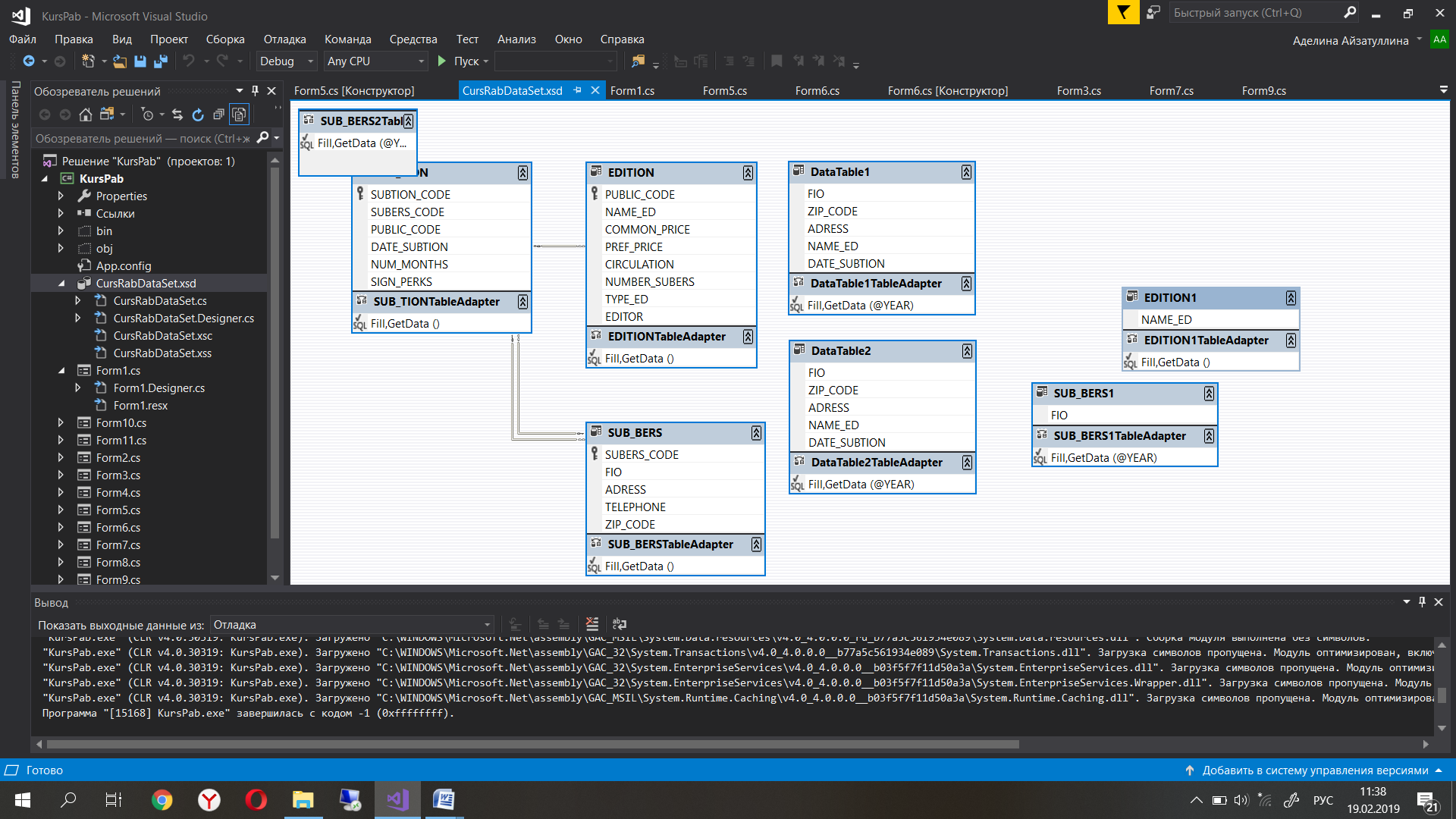
AND NOT EXISTS( SELECT \* FROM SUB\_TION

WHERE (SUB\_BERS.SUBERS\_CODE = SUB\_TION.SUBERS\_CODE)

AND (EDITION.PUBLIC\_CODE = SUB\_TION.PUBLIC\_CODE)

AND (YEAR(SUB\_TION.DATE\_SUBTION) = 2015))

);



Запрос при помощи SQL:

using System;

…

namespace KursPab

{

public partial class Form8 : Form

{

public Form8()

{

InitializeComponent();

}

private void Form8\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION1". При необходимости она может быть перемещена или удалена.

this.eDITION1TableAdapter.Fill(this.cursRabDataSet.EDITION1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_TION". При необходимости она может быть перемещена или удалена.

this.sUB\_TIONTableAdapter.Fill(this.cursRabDataSet.SUB\_TION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_BERS". При необходимости она может быть перемещена или удалена.

this.sUB\_BERSTableAdapter.Fill(this.cursRabDataSet.SUB\_BERS);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION". При необходимости она может быть перемещена или удалена.

this.eDITIONTableAdapter.Fill(this.cursRabDataSet.EDITION);

dataGridView1.AutoGenerateColumns = true;

this.eDITION1TableAdapter.Fill(cursRabDataSet.EDITION1);

this.Text = "Запрос на основании объекта TableAdapter";

}

}

}

Запрос при помощи циклов:

using System;

…

namespace KursPab

{

public partial class Form9 : Form

{

public Form9()

{

InitializeComponent();

}

private void Form9\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION1". При необходимости она может быть перемещена или удалена.

this.eDITION1TableAdapter.Fill(this.cursRabDataSet.EDITION1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_TION". При необходимости она может быть перемещена или удалена.

this.sUB\_TIONTableAdapter.Fill(this.cursRabDataSet.SUB\_TION);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.SUB\_BERS". При необходимости она может быть перемещена или удалена.

this.sUB\_BERSTableAdapter.Fill(this.cursRabDataSet.SUB\_BERS);

// TODO: данная строка кода позволяет загрузить данные в таблицу "cursRabDataSet.EDITION". При необходимости она может быть перемещена или удалена.

this.eDITIONTableAdapter.Fill(this.cursRabDataSet.EDITION);

this.Text = " Запрос, содержащий подзапросы (Record)";

dataGridView1.AutoGenerateColumns = true;

zapr();

}

public void zapr()

{

cursRabDataSet.EDITION1.Clear();

bool aE, eS;

aE = false;

foreach (CursRabDataSet.EDITIONRow eRow in cursRabDataSet.EDITION.Rows)

{

if (eRow.TYPE\_ED == "газета")

{

aE = true;

foreach (CursRabDataSet.SUB\_BERSRow rRow in cursRabDataSet.SUB\_BERS.Rows)

{

aE = (rRow.ADRESS.Contains("Казань"));

if (aE)

{

eS = false;

foreach (CursRabDataSet.SUB\_TIONRow sRow in cursRabDataSet.SUB\_TION.Rows)

{

eS = ((sRow.DATE\_SUBTION >= Convert.ToDateTime("01.01.2015")) && (sRow.DATE\_SUBTION <= Convert.ToDateTime("31.12.2015"))

&& (eRow.PUBLIC\_CODE == sRow.PUBLIC\_CODE) && (rRow.SUBERS\_CODE == sRow.SUBERS\_CODE));

if (eS) break;

}

aE = eS;

}

if (!aE) break;

}

if (aE)

{

CursRabDataSet.EDITION1Row zRow = cursRabDataSet.EDITION1.NewEDITION1Row();

zRow.NAME\_ED = eRow.NAME\_ED;

cursRabDataSet.EDITION1.Rows.Add(zRow);

}

}

}

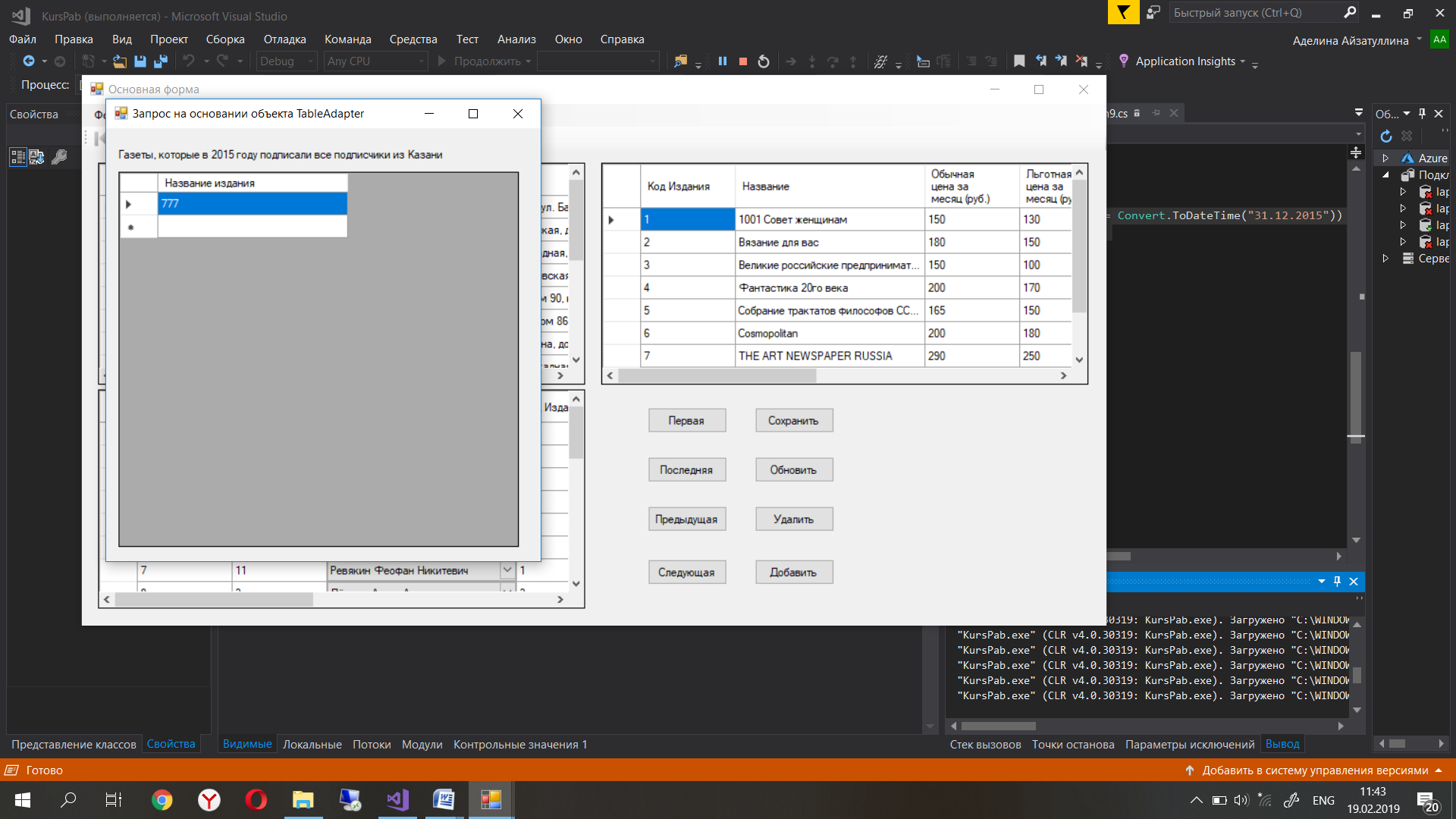
dataGridView1.Refresh();

}

}

}

Результат:



3) Для каждого подписчика найти и вывести:

* ФИО
* количество газет и количество журналов, на которые он подписался в заданном году (год является параметром)

SELECT FIO, COUNT(CASE WHEN EDITION.TYPE\_ED='журнал' THEN 1 ELSE NULL END)

AS COUNT\_M,

COUNT (CASE WHEN EDITION.TYPE\_ED = 'газета' THEN 1 ELSE NULL END) AS COUNT\_N

FROM SUB\_BERS, SUB\_TION, EDITION

WHERE

SUB\_BERS.SUBERS\_CODE = SUB\_TION.SUBERS\_CODE AND

SUB\_TION.PUBLIC\_CODE = EDITION.PUBLIC\_CODE

AND YEAR(SUB\_TION.DATE\_SUBTION) = @YEAR

GROUP BY SUB\_BERS.FIO;

using System;

…

namespace KursPab

{

public partial class Form11 : Form

{

public Form11()

{

InitializeComponent();

}

private void Form11\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

int g = 0, kod = -1, j = 0;

cursRabDataSet.SUB\_BERS2.Clear();

CursRabDataSet.SUB\_BERS2Row zRow = null;

foreach (CursRabDataSet.SUB\_TIONRow rRow in cursRabDataSet.SUB\_TION.Rows)

{

if (rRow.DATE\_SUBTION.Year.ToString() == Convert.ToString(textBox1.Text))

{

CursRabDataSet.EDITIONRow tRow = cursRabDataSet.EDITION.Rows.Find(rRow.PUBLIC\_CODE) as CursRabDataSet.EDITIONRow;

CursRabDataSet.SUB\_BERSRow oRow = cursRabDataSet.SUB\_BERS.Rows.Find(rRow.SUBERS\_CODE) as CursRabDataSet.SUB\_BERSRow;

if (zRow == null)

{

kod = rRow.SUBERS\_CODE;

if (tRow.TYPE\_ED == "журнал")

j = 1;

if (tRow.TYPE\_ED == "газета")

g = 1;

zRow = cursRabDataSet.SUB\_BERS2.NewSUB\_BERS2Row();

}

else

if (kod == rRow.SUBERS\_CODE)

{

if (tRow.TYPE\_ED == "журнал")

j++;

if (tRow.TYPE\_ED == "газета")

g++;

}

else

{

zRow.FIO = cursRabDataSet.SUB\_BERS.FindBySUBERS\_CODE(kod).FIO;

zRow.COUNT\_M = j;

zRow.COUNT\_N = g;

cursRabDataSet.SUB\_BERS2.Rows.Add(zRow);

kod = rRow.SUBERS\_CODE;//подготовливаем новую строку

if (tRow.TYPE\_ED == "журнал")

{ j = 1; g = 0; }

if (tRow.TYPE\_ED == "газета")

{ g = 1; j = 0; }

zRow = cursRabDataSet.SUB\_BERS2.NewSUB\_BERS2Row();

}

}

}

if (zRow != null)

{

//создали новую строку таблицы «Zadacha1»

zRow.COUNT\_M = j;

zRow.COUNT\_N = g;

zRow.FIO = cursRabDataSet.SUB\_BERS.FindBySUBERS\_CODE(kod).FIO;

cursRabDataSet.SUB\_BERS2.Rows.Add(zRow);

dataGridView2.Refresh();

this.Text = "Запрос через поиск по ключу";

}

}

}

}

Результат:

