МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра технічної кібернетики

**ЗВІТ**

**ДО КОМП’ЮТЕРНОГО ПРАКТИКУМУ**

**З КРЕДИТНОГО МОДУЛЯ**

**“МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ ПРОГРАМНОГО**

**ЗАБЕЗПЕЧЕННЯ”**

Виконав студент

групи ЗПІ-ЗП61

Макіян С.А.

Перевірив

К.т.н., доц. Мелкумян Е.Ю.

Київ – 2018

# Назва та опис елементів ІС «Телефона станція»

Система працює з наступними елементами даних

* Номер абонента, пільги на абонплату, адреса, дата установки телефону, наявність блокатора, кількість апаратів, примітки
* Розмір абонплати, сума платежу за кількість хвилин по місту (кількість хвилин, плата за хвилину), сума платежів за міжміські переговори (кількість хвилин за переговори з певним пунктом, вартість хвилини), дата поступлення платежу, заборгованість, сума заборгованості

# Проектування моделі предметної області

2.1 перелік екторів та прецендетів

Табл. 1. Перелік екторів та прецедентів

|  |  |
| --- | --- |
| Клієнт | Робить дзвінки, отримує рахунок, платить абонплату, може запросити статистику |
| Касир | Здіснює платіж по абонплати |
| Адміністратор | Редагує базу даних |
| Банківська підсистема | Проводить оплату абонплати  Підтверджує оплату для подальшої видачі білета |

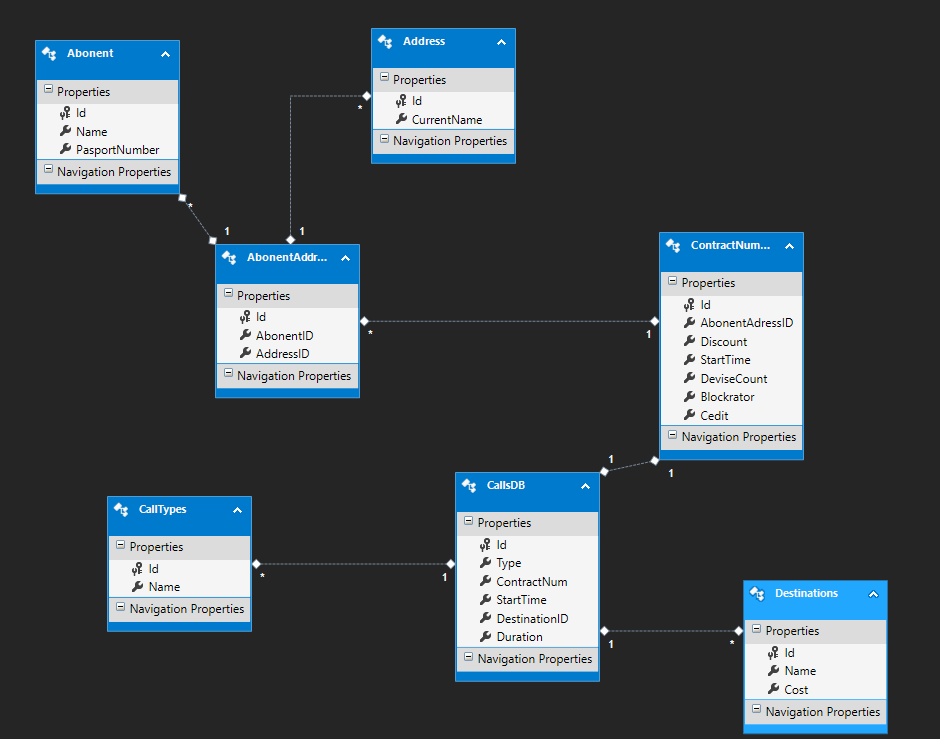
2.2 Діаграма процендентів



# Інфологічне проектування системи

3.1 Використання методології DFD

 3.2 Використання методології сутність зв’язок



Для побудови ER моделі було ми використали я в проекті з Windows Form створив Model ADO.NET EDM

Далі генеруємо на основі УК моделі за допомогою ORM EF (V6) БД, ООП класи з сутностей.

namespace Smtirpz

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class Model1Container : DbContext

{

public Model1Container()

: base("name=Model1Container")

{

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public virtual DbSet<Abonent> Abonents { get; set; }

public virtual DbSet<Address> Addresses { get; set; }

public virtual DbSet<AbonentAddress> AbonentAddresses { get; set; }

public virtual DbSet<ContractNumber> ContractNumbers { get; set; }

public virtual DbSet<CallsDB> CallsDBs { get; set; }

public virtual DbSet<CallTypes> CallTypes { get; set; }

public virtual DbSet<Destinations> Destinations { get; set; }

}

}

namespace Smtirpz

{

using System;

using System.Collections.Generic;

public partial class Abonent

{

public int Id { get; set; }

public string Name { get; set; }

public string PasportNumber { get; set; }

}

}

namespace Smtirpz

{

using System;

using System.Collections.Generic;

public partial class Address

{

public int Id { get; set; }

public string CurrentName { get; set; }

}

}

namespace Smtirpz

{

using System;

using System.Collections.Generic;

public partial class AbonentAddress

{

public int Id { get; set; }

public int AbonentID { get; set; }

public int AddressID { get; set; }

}

}

namespace Smtirpz

{

using System;

using System.Collections.Generic;

public partial class CallsDB

{

public int Id { get; set; }

public int Type { get; set; }

public int ContractNum { get; set; }

public System.TimeSpan StartTime { get; set; }

public int DestinationID { get; set; }

public long Duration { get; set; }

}

}

namespace Smtirpz

{

using System;

using System.Collections.Generic;

public partial class CallTypes

{

public int Id { get; set; }

public string Name { get; set; }

}

}

namespace Smtirpz

{

using System;

using System.Collections.Generic;

public partial class CallTypes

{

public int Id { get; set; }

public string Name { get; set; }

}

}

namespace Smtirpz

{

using System;

using System.Collections.Generic;

public partial class Destinations

{

public int Id { get; set; }

public string Name { get; set; }

public double Cost { get; set; }

}

}

SQL код для генерації бази даних

-- --------------------------------------------------

-- Entity Designer DDL Script for SQL Server 2005, 2008, 2012 and Azure

-- --------------------------------------------------

-- Date Created: 02/23/2018 07:55:30

-- Generated from EDMX file: c:\users\sim\documents\visual studio 2015\Projects\Smtirpz\Smtirpz\Model1.edmx

-- --------------------------------------------------

SET QUOTED\_IDENTIFIER OFF;

GO

USE [SMTirpz];

GO

IF SCHEMA\_ID(N'dbo') IS NULL EXECUTE(N'CREATE SCHEMA [dbo]');

GO

-- --------------------------------------------------

-- Dropping existing FOREIGN KEY constraints

-- --------------------------------------------------

-- --------------------------------------------------

-- Dropping existing tables

-- --------------------------------------------------

-- --------------------------------------------------

-- Creating all tables

-- --------------------------------------------------

-- Creating table 'Abonents'

CREATE TABLE [dbo].[Abonents] (

[Id] int IDENTITY(1,1) NOT NULL,

[Name] nvarchar(max) NOT NULL,

[PasportNumber] nvarchar(max) NOT NULL

);

GO

-- Creating table 'Addresses'

CREATE TABLE [dbo].[Addresses] (

[Id] int IDENTITY(1,1) NOT NULL,

[CurrentName] nvarchar(max) NOT NULL

);

GO

-- Creating table 'AbonentAddresses'

CREATE TABLE [dbo].[AbonentAddresses] (

[Id] int IDENTITY(1,1) NOT NULL,

[AbonentID] int NOT NULL,

[AddressID] int NOT NULL

);

GO

-- Creating table 'ContractNumbers'

CREATE TABLE [dbo].[ContractNumbers] (

[Id] int IDENTITY(1,1) NOT NULL,

[AbonentAdressID] int NOT NULL,

[Discount] float NOT NULL,

[StartTime] datetime NOT NULL,

[DeviseCount] int NOT NULL,

[Blockrator] int NOT NULL,

[Cedit] float NOT NULL

);

GO

-- Creating table 'CallsDBs'

CREATE TABLE [dbo].[CallsDBs] (

[Id] int IDENTITY(1,1) NOT NULL,

[Type] int NOT NULL,

[ContractNum] int NOT NULL,

[StartTime] time NOT NULL,

[DestinationID] int NOT NULL,

[Duration] bigint NOT NULL

);

GO

-- Creating table 'CallTypes'

CREATE TABLE [dbo].[CallTypes] (

[Id] int IDENTITY(1,1) NOT NULL,

[Name] nvarchar(max) NOT NULL

);

GO

-- Creating table 'Destinations'

CREATE TABLE [dbo].[Destinations] (

[Id] int IDENTITY(1,1) NOT NULL,

[Name] nvarchar(max) NOT NULL,

[Cost] float NOT NULL

);

GO

-- --------------------------------------------------

-- Creating all PRIMARY KEY constraints

-- --------------------------------------------------

-- Creating primary key on [Id] in table 'Abonents'

ALTER TABLE [dbo].[Abonents]

ADD CONSTRAINT [PK\_Abonents]

PRIMARY KEY CLUSTERED ([Id] ASC);

GO

-- Creating primary key on [Id] in table 'Addresses'

ALTER TABLE [dbo].[Addresses]

ADD CONSTRAINT [PK\_Addresses]

PRIMARY KEY CLUSTERED ([Id] ASC);

GO

-- Creating primary key on [Id] in table 'AbonentAddresses'

ALTER TABLE [dbo].[AbonentAddresses]

ADD CONSTRAINT [PK\_AbonentAddresses]

PRIMARY KEY CLUSTERED ([Id] ASC);

GO

-- Creating primary key on [Id] in table 'ContractNumbers'

ALTER TABLE [dbo].[ContractNumbers]

ADD CONSTRAINT [PK\_ContractNumbers]

PRIMARY KEY CLUSTERED ([Id] ASC);

GO

-- Creating primary key on [Id] in table 'CallsDBs'

ALTER TABLE [dbo].[CallsDBs]

ADD CONSTRAINT [PK\_CallsDBs]

PRIMARY KEY CLUSTERED ([Id] ASC);

GO

-- Creating primary key on [Id] in table 'CallTypes'

ALTER TABLE [dbo].[CallTypes]

ADD CONSTRAINT [PK\_CallTypes]

PRIMARY KEY CLUSTERED ([Id] ASC);

GO

-- Creating primary key on [Id] in table 'Destinations'

ALTER TABLE [dbo].[Destinations]

ADD CONSTRAINT [PK\_Destinations]

PRIMARY KEY CLUSTERED ([Id] ASC);

GO

-- --------------------------------------------------

-- Creating all FOREIGN KEY constraints

-- --------------------------------------------------

-- Creating foreign key on [AbonentID] in table 'AbonentAddresses'

ALTER TABLE [dbo].[AbonentAddresses]

ADD CONSTRAINT [FK\_AbonentAbonentAddress]

FOREIGN KEY ([AbonentID])

REFERENCES [dbo].[Abonents]

([Id])

ON DELETE NO ACTION ON UPDATE NO ACTION;

GO

-- Creating non-clustered index for FOREIGN KEY 'FK\_AbonentAbonentAddress'

CREATE INDEX [IX\_FK\_AbonentAbonentAddress]

ON [dbo].[AbonentAddresses]

([AbonentID]);

GO

-- Creating foreign key on [AddressID] in table 'AbonentAddresses'

ALTER TABLE [dbo].[AbonentAddresses]

ADD CONSTRAINT [FK\_AddressAbonentAddress]

FOREIGN KEY ([AddressID])

REFERENCES [dbo].[Addresses]

([Id])

ON DELETE NO ACTION ON UPDATE NO ACTION;

GO

-- Creating non-clustered index for FOREIGN KEY 'FK\_AddressAbonentAddress'

CREATE INDEX [IX\_FK\_AddressAbonentAddress]

ON [dbo].[AbonentAddresses]

([AddressID]);

GO

-- Creating foreign key on [AbonentAdressID] in table 'ContractNumbers'

ALTER TABLE [dbo].[ContractNumbers]

ADD CONSTRAINT [FK\_AbonentAddressContractNumber]

FOREIGN KEY ([AbonentAdressID])

REFERENCES [dbo].[AbonentAddresses]

([Id])

ON DELETE NO ACTION ON UPDATE NO ACTION;

GO

-- Creating non-clustered index for FOREIGN KEY 'FK\_AbonentAddressContractNumber'

CREATE INDEX [IX\_FK\_AbonentAddressContractNumber]

ON [dbo].[ContractNumbers]

([AbonentAdressID]);

GO

-- Creating foreign key on [Type] in table 'CallsDBs'

ALTER TABLE [dbo].[CallsDBs]

ADD CONSTRAINT [FK\_CallTypesCallsDB]

FOREIGN KEY ([Type])

REFERENCES [dbo].[CallTypes]

([Id])

ON DELETE NO ACTION ON UPDATE NO ACTION;

GO

-- Creating non-clustered index for FOREIGN KEY 'FK\_CallTypesCallsDB'

CREATE INDEX [IX\_FK\_CallTypesCallsDB]

ON [dbo].[CallsDBs]

([Type]);

GO

-- Creating foreign key on [DestinationID] in table 'CallsDBs'

ALTER TABLE [dbo].[CallsDBs]

ADD CONSTRAINT [FK\_DestinationsCallsDB]

FOREIGN KEY ([DestinationID])

REFERENCES [dbo].[Destinations]

([Id])

ON DELETE NO ACTION ON UPDATE NO ACTION;

GO

-- Creating non-clustered index for FOREIGN KEY 'FK\_DestinationsCallsDB'

CREATE INDEX [IX\_FK\_DestinationsCallsDB]

ON [dbo].[CallsDBs]

([DestinationID]);

GO

-- Creating foreign key on [ContractNum] in table 'CallsDBs'

ALTER TABLE [dbo].[CallsDBs]

ADD CONSTRAINT [FK\_ContractNumberCallsDB]

FOREIGN KEY ([ContractNum])

REFERENCES [dbo].[ContractNumbers]

([Id])

ON DELETE NO ACTION ON UPDATE NO ACTION;

GO

-- Creating non-clustered index for FOREIGN KEY 'FK\_ContractNumberCallsDB'

CREATE INDEX [IX\_FK\_ContractNumberCallsDB]

ON [dbo].[CallsDBs]

([ContractNum]);

GO

-- --------------------------------------------------

-- Script has ended

-- --------------------------------------------------

# Діаграма класів

Клас Abonent містить інформацію про клієнта: його номер, адресу, дату реєстрації тощо. Він пов’язен ассоціцією з классом Bank, та породжує об’єкти класу Call. В классы Call є інформація про початок дзвінка, номер абонента, тривалість дзвінка, напрямок та його ціна. Він заноситься в клас Db, котрий породжує клас Bill. В класі Bill є інформація про розмір абонплати та номер клієнта. Класс Cashier зв’язан ассоціацієй з классом Bank, Bill та Abonent та відповідає за проведення Bill через Bank. На виході Abonent отримує Bill, що пройшов через Bank.

# Моделювання поведінки системи

Діаграма активності



Діаграма послідовності

