МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ В.Н.КАРАЗІНА

ФАКУЛЬТЕТ МАТЕМАТКИ ТА ІНФОРМАТИКИ

КАФЕДРА ТЕОРЕТИЧНОЇ І ПРИКЛАДНОЇ ІНФОРМАТИКИ

КВАЛІФІКАЦІЙНА РОБОТА

ТЕМА: «**Розробка архітектури серверної частини інформаційних систем «Відділ контрактної освіти» і  «Центр міжнародної освіти»»**

Виконавець: студент VI курсу, групи МФ-61

напряму підготовки 7.040302 – інформатика

Башкатов Д.І.

Науковий керівник: Морозова А.Г.

Рецензент: Зарецька І.Т.

Харків – 2017

**Зміст**

[РОЗДІЛ 1. ВСТУП 3](#_Toc483170340)

[1.1 Формулювання задачі, мета і предмет проекту, обґрунтування актуальності теми 3](#_Toc483170341)

[1.2 Учасники проекту 3](#_Toc483170342)

[1.3 Попередні технічні рішення 3](#_Toc483170343)

[1.4. Короткі відомості про отримані результати 3](#_Toc483170344)

[РОЗДІЛ 2. ОСНОВНА ЧАСТИНА 3](#_Toc483170345)

[2.1 Вимоги до системи 3](#_Toc483170346)

[2.1.1 Рамки системи 3](#_Toc483170347)

[2.1.2 Вимоги до предметної області 3](#_Toc483170348)

[2.1.3 Вимоги до даних 3](#_Toc483170349)

[2.1.4 Інші вимоги 3](#_Toc483170350)

[2.3 Загальна постановка задачі 3](#_Toc483170351)

[2.4 Опис предметної області 3](#_Toc483170352)

[2.5 Опис функціональності системи 3](#_Toc483170353)

[2.6 Архітектура системи 3](#_Toc483170354)

[2.7 Використовувані технології 3](#_Toc483170356)

[2.8 Використані шаблони проектування 3](#_Toc483170357)

[2.9 Розробка діаграми пакетів. 3](#_Toc483170358)

[2.10 Опис нововведень 3](#_Toc483170359)

[2.10.1 Розширення Бази Даних 3](#_Toc483170360)

[2.10.2 Розширення віддалених методів (RMI) 3](#_Toc483170361)

[РОЗДІЛ 3. ВИСНОВОК 3](#_Toc483170362)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 3](#_Toc483170363)

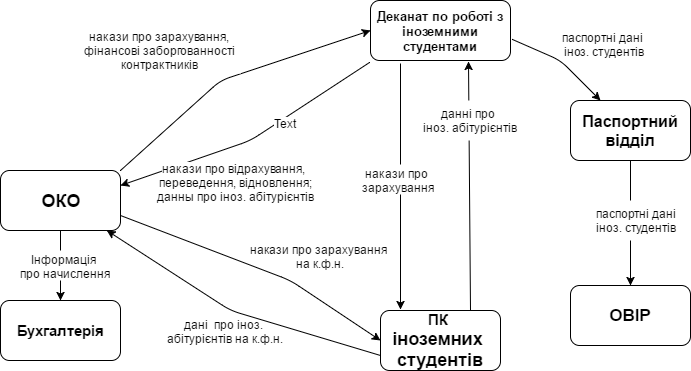
## РОЗДІЛ 1. ВСТУП

У Харківському національному університеті імені В.Н. Каразіна ведеться автоматизація всіх процесів. Прикладом цього може слугувати інформаційна система «ВКО», яка автоматизує процес документообігу відділу контрактної освіти (ВКО) та Центру міжнародної освіти (ЦМО) університету. Актуальною стала задача розширення і оновлення існуючої системи «ВКО».

Центр міжнародної освіти включає в себе три відділи, що взаємодіють один з одним: деканат по роботі з іноземними студентами, приймальна комісія іноземних студентів і паспортний відділ. Також Центр міжнародної освіти взаємодіє з відділом контрактної освіти. Всі вищеназвані відділи об'єднує одне - робота з іноземними студентами та абітурієнтами ХНУ. За всіма відділами закріплені певні повноваження. Деякі повноваження є унікальними для кожного відділу, а деякі - пересічними для кількох відділів. Крім того, ці відділи постійно взаємодіють один з одним, обмінюються даними, доповнюють їх. Відбувається процес активного документообігу між відділами. Виникають ситуації, коли одному відділу для подальшої роботи необхідні дані іншого відділу.

Тому актуальною стає завдання реінжинірингу існуючої системи для забезпечення роботи системи на сучасних системах, а також впровадження нових операцій з нарахуваннями, наказами та пенею за несвоєчасну оплату, введену цього року. З тих пір, як поточна версія системи «ВКО» створювалася (а пізніше модернізувалася) пройшло кілька років і сьогодні система не оптимізована під сучасні платформи.

**ВКО**



*Мал. 1. Схема взаємодії відділів університету.*

Відділ контрактного навчання взаємодіє з усім комплексом центру міжнародної освіти. ВКО забезпечує ЦМО даними про іноземних студентів і абітурієнтів контрактної форми навчання, наказами про їх зарахування, фінансової заборгованості контрактників. ЦМО забезпечує ВКО паспортними даними студентів контрактної форми навчання, наказами про відрахування, переведення, поновлення студентів та абітурієнтів контрактної форми навчання.

ВКО взаємодіє з бухгалтерією університету. Бухгалтерію цікавить інформація про оплати студентів і абітурієнтів контрактної форми навчання.

Паспортні дані іноземних студентів надходять в ОВІР.

Відділи центру міжнародної освіти взаємодіють між собою дуже сильно. Це проявляється в обміні даними іноземних студентів, документами різних типів: про зарахування, переведення, поновлення, довідками (наприклад, «про місце навчання»).

Центр міжнародної освіти контролює дані про іноземних студентів, як контрактної форми навчання, так і бюджетної.

Центр міжнародної освіти обмінюється даними з відділом контрактного навчання. Сфери їх діяльності перехрещуються, так як відділ контрактного навчання виробляє роботу зі студентами контрактної форми навчання. Студентами контрактної форми навчання є громадяни України та іноземні піддані.

Таким чином, автоматизація спільних процесів центру міжнародної освіти та відділу контрактного навчання є актуальним завданням.

Відділ контрактного навчання і відділи Центру міжнародної освіти вже оснащені автоматизованою системою «ВКО», яка дозволяє контролювати потоки даних і інформації, що стосуються кожного з відділів. Також існує загальна база даних, яка зберігає інформацію про іноземних студентів та студентів громадян України контрактної форми навчання.

На основі проведеного аналізу було прийнято рішення про реинжиниринг автоматизованої системи «ВКО».

## Формулювання задачі, мета і предмет проекту, обґрунтування актуальності теми

Був проведений ретельний аналіз попередньої версії системи. При аналізі предметної області було виявлено, що дана задача є досить актуальною, тому що в одній з основних завдань контрактного відділу є робота з оплатами та нарахувань. На той момент весь цей процес не був автоматизованим і кожен місяць оператору доводиться вносити інформацію, пов'язану з оплатами і нарахуваннями. Також з цього року була введена пеня за заборгованість за контрактом. У зв'язку з цим з'явилося необхідність розширити поточну систему.

Крім того при аналізі було виявлено, що система сильно застаріла, тому що створювалася в 2004 році.

У зв'язку з цим була розроблена наступна постановка завдання і мета проекту.

Постановка задачі:

Здійснення реінжинірингу існуючої інформаційної системи «ВКО» для забезпечення роботи системи на сучасних системах, розширення роботи системи з нарахуваннями, додавання функції нарахування пені за несвоєчасну оплату, а також автоматизація роботи з наказами. Також було вирішенно добавити до функціональності системи засоби логування роботи для контролю і відстежиння.

Таким чином, виходячи з актуальності проекту, було прийнято рішення про оновлення системи.

## 1.2 Учасники проекту

Замовник: Центр міжнародної освіти та відділ контрактної освіти Харківського національного університету імені В.Н.Каразіна.

Виконавець: Студент факультету математики та інформатики групи МФ-61 під керівництвом Морозової А.Г.

## 1.3 Попередні технічні рішення

«ВКО» повинен бути мережевим, переносних незалежним програмним комплексом. Використовувані при розробці програмні засоби повинні поширюватися виробниками безкоштовно і без обмежень щодо їх використання.

База даних повинна містити інформацію про абітурієнтів та студентів контрактної форми навчання, вироблених ними оплатах, іноземних студентах контрактної та бюджетної форм навчання, паспортних даних іноземних студентів, а також всю інформацію, що проходить через деканат по роботі з іноземними студентами, приймальної комісії, паспортний відділ і відділ контрактного навчання.

Доступ до бази даних системи повинен бути розрахованих на багато користувачів.

Весь існуючий функціонал повинен збережеться.

## 1.4. Короткі відомості про отримані результати

Виконана в рамках проекту робота спрямована на вирішення актуальної задачі реінжинірингу існуючої інформаційної системи «ВКО», яка автоматизує процес роботи з нарахуваннями, вносить зміни в роботу наказів і додає в систему нарахування пені. Отримані результати мають велике практичне значення для вирішення завдання створення автоматизованих інформаційних систем, які дозволяють здійснювати контроль процесів обміну даними і документообігу.

В процесі виконання роботи отримані практичні результати, які полягають в наступному:

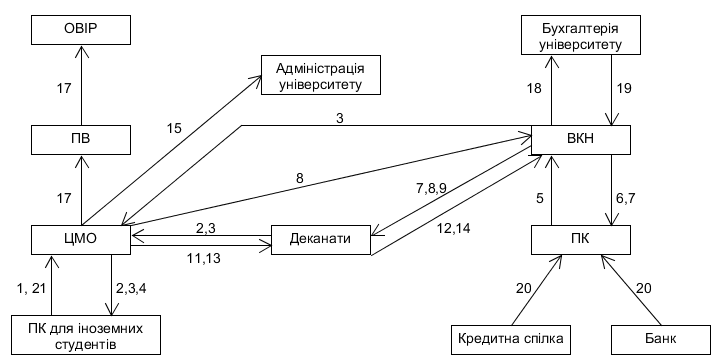
● На основі проведеного аналізу була оновлена ​​серверна частина системи «ВКО».

# РОЗДІЛ 2. ОСНОВНА ЧАСТИНА

## 2.1 Вимоги до системи

### 2.1.1 Рамки системи

Рамки системи визначаються відповідно до наведеної нижче діаграми контексту (Мал.1), яка визначає взаємодію з основними зовнішніми об'єктами.



*Мал.2. Діаграма контексту*

Пояснення до діаграми: ЦМО - центр міжнародної освіти, ВКН - відділ контрактного навчання, ПВ - паспортний відділ, ПК - приймальна комісія.

**Потоки:**

**1** – дані про іноземних абітурієнтів, які успішно склали вступні іспити на контрактну форму навчання або місця державного замовлення;

**2** – проект наказу про зарахування іноземних абітурієнтів на контрактну форму навчання або місця державного замовлення;

**3** – наказ про зарахування іноземних абітурієнтів на контрактну форму навчання;

**4** – наказ про зарахування іноземних абітурієнтів на бюджетну форму навчання;

**5** – дані про абітурієнтів гр. Україна, які успішно склали вступні іспити на контрактну форму навчання;

**6** – проект наказу про зарахування абітурієнтів гр. Україна на контрактну форму навчання;

**7** – наказ про зарахування абітурієнтів гр. Україна на контрактну форму навчання;

**8** – дані про іноземних абітурієнтів;

**9** – дані про фінансові заборгованості іноземних студентів.;

**10** – дані про фінансові заборгованості студентів гр. України;

**11** – накази про відрахування, переведення на наступний курс, відновлення іноземних студентів на контрактну форму навчання або місця державного замовлення;

**12** – накази про відрахування, переведення на наступний курс, відновлення студентів гр. Україна на контрактну форму навчання;

**13** – дані про іноземних студентів дистанційної форми навчання;

**14** – дані про студентів гр. Україна дистанційної форми навчання;

**15** – звіти про поточні дані іноземних студентів на контрактну форму навчання або місця державного замовлення;

**16** – звіти про поточні дані студентів гр. Україна на контрактну форму навчання;

**17** – паспортні дані іноземних студентів;

**18** – інформація про нарахування;

**19** – інформація про оплати по перерахуванню;

**20** – дані про оплати через банк і Кредитний союз;

**21** – дані про нострифікацію.

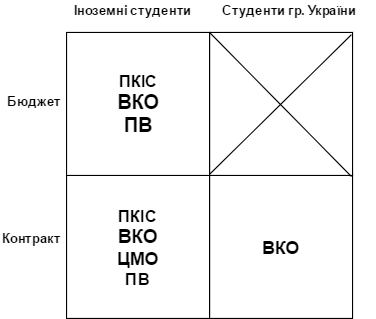
Дана система складається з чотирьох підсистем: деканату по роботі з іноземними студентами, приймальної комісії для іноземних студентів, паспортного відділу - підсистеми, що відносяться до відділів ЦМО; і відділу контрактного навчання. Система забезпечує роботу чотирьох підсистем з єдиною базою даних.

Область діяльності підсистем можна проілюструвати наступною схемою:

**ВКО**

**ВКО**

**ВКО**



Даними іноземних студентів і абітурієнтів бюджетної форми навчання займаються:

- приймальна комісія для іноземних студентів;

- центр міжнародної освіти;

- паспортний відділ.

Даними іноземних студентів і абітурієнтів контрактної форми навчання займаються:

- приймальна комісія для іноземних студентів;

- центр міжнародної освіти;

- відділ контрактного навчання;

- паспортний відділ.

З даними студентів і абітурієнтів громадян України контрактної форми навчання працює відділ контрактного навчання.

Робота з даними студентів і абітурієнтів громадян України бюджетної форми навчання не провадиться.

Розробляється система також повинна надавати можливість зміни статусу учня (абітурієнт, студент, відрахований студент і т.д.).

**2.1.2 Вимоги до предметної області**

На етапі планування системи були виділені наступні вимоги предметної області:

Робота з особистими даними про студентів і абітурієнтів:

* Реєстрація інформації про абітурієнтів;
* Реєстрація інформації про студентів;
* Пошук інформації про абітурієнтів;
* Пошук інформації про студентів;
* Редагування інформації про абітурієнтів;
* Редагування інформації про студентів;
* Видалення інформації про абітурієнтів;
* Видалення інформації про студентів;
* Друк інформації про абітурієнтів;
* Друк інформації про студентів.

Робота з паспортними даними про іноземних студентів і абітурієнтів:

* Реєстрація інформації про паспортні дані абітурієнтів;
* Редагування інформації про паспортні дані абітурієнтів;
* Редагування інформації про паспортні дані студентів.

Робота з довідками про іноземних студентів і абітурієнтів:

* Друк довідок про іноземні абітурієнтів;
* Друк довідок про іноземних студентів.

Робота з наказами.

* Створення проекту наказу про зарахування;
* Створення проекту наказу на відрахування;
* Створення проекту наказу про переведення на наступний курс;
* Створення наказу про відновлення;
* Реєстрація наказів;
* Пошук по наказам;
* Друк наказів;
* Створення проекту інших наказів;
* Перехід на наступний курс за наказом.

Робота зі звітами:

* Робота з генератором звітів;
* Робота з базою звітів.

Робота з оплатами студентів і абітурієнтів на к.ф.н.:

* Додавання даних про оплату;
* Видалення даних про оплату;
* Розрахунок суми, що підлягає;
* Друк даних про оплату;
* Пошук даних про оплати;

Створення нарахувань на місяць.

Нарахування пені.

Робота з редактором оплат.

Пошук студентів з невірними підсумками.

**2.1.3 Вимоги до даних**

Інформація про студента повинна включати:

* прізвище, ім'я, по батькові;
* громадянство (країна);
* факультет, напрямок і спеціальність надходження;
* фінансова одиниця, за якою буде значитися студент;
* номер наказу про зарахування та дата зарахування;
* дата початку та дата планованого закінчення періоду навчання;
* форма навчання;
* освітньо-кваліфікаційний рівень;
* освітня програма;
* поточний курс навчання, загальна кількість курсів;
* номер і сума контракту (суми щорічної оплати)
* тип контракту;
* тип платника;
* дані про платника, якщо це юридична особа;
* лінія прибуття студента;
* регіон;
* мова навчання;
* дата народження;
* дата видачі паспорта;
* термін дії паспорта;
* підлогу;
* паспорт №;
* реєстрація №;
* термін дії реєстрації №;
* місце колишньої реєстрації;
* віза №;
* ким видана віза;
* дата видачі віза;
* термін дії візи;
* межа (пункт перетину кордону);
* дата прибуття;
* адреса проживання;
* дата зарахування.

Інформація про контракт має включати:

* тип контракту (валюта);
* дата здійснення оплати за контрактом;
* сума контракту;
* номер контракту.

Інформація про накази повинна включати:

* номер наказу;
* дата реєстрації наказу;
* дата виконання;
* тип наказу (зарахування, переведення, відрахування).

Інформація про оплату повинна включати:

* номер транзакції;
* сума до оплати;
* дата оплати;
* причина оплати;
* спосіб здійснення оплати (оплата через касу, перерахування тощо);
* банк, у якому відбувалася оплата;
* платник;
* навчальний рік, в якому здійснено оплату.

### 2.1.4 Інші вимоги

Вимоги до інтерфейсу:

Кінцеві користувачі «ВКО» повинні взаємодіяти з системою, використовуючи інтерфейс типу «тонкий клієнт». Час навчання співробітників відділів роботі з інтерфейсом системи не повинно перевищувати 2 тижнів. Інтерфейс повинен бути зрозумілим і доступним співробітникам всіх відділів.

Вимоги до безпеки інформації:

Всі користувачі системи «ВКО» (оператор, адміністратор) повинні мати різні права доступу, регульовані відповідно до спеціально розробленої політикою безпеки. Система повинна забезпечувати можливість додавання нових операторів.

Для отримання доступу до системи адміністратор повинен ввести пароль. Адміністратор має доступ до функцій модифікації інформації.

Вимоги до захищеності інформації:

Робочі місця співробітників деканату, ВКО, ПО, ПК для іноземних абітурієнтів повинні бути з'єднані локальною мережею і не мати доступу до мережі Інтернет. Для забезпечення цілісності та конфіденційності даних комп'ютери співробітників повинні не мати таких пристроїв читання-запису інформації, як: DVD-ROM, CD-ROM.

Вимоги до доступності інформації:

Повна документація по розробці проекту і вихідний код є вільно поширюваними.

Вимоги до переносимості:

Система призначена для функціонування в операційному середовищі Windows і Linux. Переносимість на інші платформи не передбачено.

Вимоги до адаптації:

Якщо відбудуться зміни в роботі будь-якого підрозділу, система повинна мати можливості щодо розширення та адаптації до нових умов.

Вимоги до ресурсів.

ВКО володіє наступними ресурсами: 5 комп'ютерів із середнім об'ємом жорсткого диска 100 ГБ і операційною системою Windows 10 і 1 комп'ютером з операційною системою Windows Server 2012.

У деканаті по роботі з іноземними студентами знаходиться 3 комп'ютери з середнім об'ємом жорсткого диска 100 ГБ і операційною системою Linux.

ПО володіє наступними ресурсами: 2 комп'ютери із середнім об'ємом жорсткого диска 100 ГБ і операційною системою Linux.

У розпорядженні ПК для іноземних абітурієнтів знаходиться 1 комп'ютер з операційною системою Windows 10 і середнім об'ємом жорсткого диска 100 ГБ.

Вимоги до стандартів.

В процесі розробки системи були враховані положення:

* статуту ХНУ ім. В.Н. Каразіна.
* статуту деканату по роботі з іноземними студентами.
* статуту відділу контрактного навчання.
* статуту приймальної комісії для іноземних студентів.
* статуту паспортного відділу.

Інші вимоги:

Система «ВКО» повинна бути реалізована з використанням тільки безкоштовного та вільно поширюваного програмного забезпечення.

**2.2** Характеристики користувачів

Система «ВКО» розрахована на користувача, який є співробітником однієї з чотирьох підсистем: деканат по роботі з іноземними студентами, відділу контрактної освіти, паспортного відділу, приймальні комісії для іноземних абітурієнтів. Розробляється система надає розмежування рівнів доступу користувачів (в залежності від типу користувача і обраної підсистеми). З системою взаємодіють 2 типу кінцевих користувачів:

1. Оператор;

2. Адміністратор.

У свою чергу функції Оператора розрізняються залежно від того, яка підсистема використовується.

Функції Оператора ПК для іноземних абітурієнтів:

* реєстрація особистих даних іноземних абітурієнтів;
* робота з особистими даними іноземних абітурієнтів;
* розширений пошук по іноземним абітурієнтам.

Функції Оператора деканату по роботі з іноземними студентами:

* робота з особистими даними іноземних абітурієнтів і студентів;
* розширений пошук по іноземним учням;
* робота з архівом;
* робота з паспортними даними іноземних абітурієнтів і студентів;
* робота з довідками іноземних абітурієнтів і студентів;
* робота з наказами іноземних абітурієнтів і студентів;
* робота зі звітами іноземних абітурієнтів і студентів.

Функції Оператора ПО:

* реєстрація паспортних даних іноземних абітурієнтів;
* робота з паспортними даними іноземних абітурієнтів і студентів;
* робота з довідками іноземних абітурієнтів і студентів.

Функції Оператора ВКО:

* реєстрація особистих даних абітурієнтів гр. Україна на к.ф.н.;
* робота з особистими даними гр. України та іноземних абітурієнтів і студентів на к.ф.н .;
* розширений пошук по гр. Україна і іноземним учням на к.ф.н ..
* робота з архівом;
* робота з наказами гр. України та іноземних абітурієнтів і студентів на к.ф.н .;
* робота зі звітами гр. України та іноземних абітурієнтів і студентів на к.ф.н .;
* робота з оплатами гр. України та іноземних абітурієнтів і студентів на к.ф.н.

Функції адміністратора включають всі функції звичайного оператора і, крім того:

* твір системних установок;
* твір нарахувань на місяць;
* робота з редактором оплат;
* пошук студентів з невірними підсумками.

## 2.3 Загальна постановка задачі

Необхідно провести реінжиніринг інформаційної системи «ВКО», яка здійснює автоматизацію документообігу відділу контрактного навчання (ВКО) і Центру міжнародної освіти (ЦМО) ХНУ імені В.Н. Каразіна, представленого деканатом по роботі з іноземними студентами, приймальною комісією для іноземних студентів і паспортним відділом.

Автоматизована система «ВКО» повинна бути спроектована так, щоб була збережена, можливість її подальшого розширення. За допомогою системи необхідно контролювати документообіг і всі процеси деканату по роботі з іноземними студентами, відділу контрактного навчання, приймальні комісії для іноземних студентів і паспортного відділу, пов'язані з іноземними студентами та абітурієнтами, студентами та абітурієнтами контрактної форми навчання. «ВКО» повинна обслуговувати одночасно всі вищевказані відділи, використовуючи одне джерело даних. Система повинна забезпечувати надійність і несуперечність даних зберігаються в базі даних.

Передбачається зберігання і супровід особистих даних студентів та абітурієнтів, в тому числі і паспортних даних. Система «ВКО» повинна дозволяти реєструвати вироблені студентами та абітурієнтами оплати, редагувати дані про оплати і нарахування. Користувач системи повинен мати можливість складання наказів, звітів та довідок, пов'язаних зі студентами та абітурієнтами та їх діяльністю.

Також повинні бути розроблені автоматизовані процеси, такі закриття навчального року та переведення студентів на наступний курс, що використовують планувальник завдань.

2.4 Опис предметної області.

Система «ВКО» є автоматизованою, що дозволяє їй зберігати інформацію необхідну деканату по роботі з іноземними студентами, приймальні комісії для іноземних студентів, паспортному відділу та відділу контрактного навчання. Сферою області відділів центру міжнародної освіти є дані іноземних студентів і абітурієнтів, їх накази, довідки та документи, пов'язані з ними. Сфера ж діяльності відділу контрактного навчання - це інформація про студентів і абітурієнтів контрактної форми навчання, накази і оплати, вироблені ними.

Всі студенти та абітурієнти мають особисті дані, які включають і паспортні дані. Для повноцінної роботи співробітникам розглянутих відділів необхідний доступ до інформації, пов'язаної зі студентами та абітурієнтами. Таким чином, «ВКО» надає доступ користувачеві до даних, що зберігаються в базі даних. Користувачами системи є співробітники вищевказаних відділів.

Користувач повністю управляє системою і даними, що зберігаються з її допомогою.

Абітурієнти та студенти контрактної форми навчання виробляють оплати, які співробітники відділу контрактного навчання (користувачі) реєструють в системі. Також користувачі системи мають можливість реєструвати накази і довідки абітурієнтів і студентів. Система «ВКО» має можливості для зберігання такої інформації.

Система дозволяє співробітникам одних відділів мати доступ до наказів, довідок і документів інших відділів.

2.5 Опис функціональності системи.

Система «ВКО» обслуговує чотири відділи університету: деканат по роботі з іноземними студентами, паспортний відділ, приймальну комісію для іноземних студентів і відділ контрактного навчання. Кожен з відділів виконує певні обов'язки. Розподіл обов'язків між відділами було враховано при розробці клієнтських місць.

Система «ВКО» складається з чотирьох підсистем:

- підсистема деканату по роботі з іноземними студентами;

- підсистема відділу контрактного навчання;

- підсистема паспортного відділу;

- підсистема приймальної комісії для іноземних студентів.

З системою взаємодіють два типи кінцевих користувачів:

- оператор

- адміністратор

Функціональність операторів підсистем була взята з попередньої версії, але була розширена, як і сама система.

У порівнянні з попередньою версією нова система «ЦМО» має нові можливості, а саме:

- робота с даними нострифікації іноземних студентів, що забезпечує можливість реєстрації і редагування даних про іноземні документи про освіту, які мають юридичну силу на території України;

- робота з посвідченнями іноземних студентів, яка складається в реєстрації документа, що посвідчує іноземного студента та підтверджує законні підстави для тимчасового проживання в Україні;

- можливість вказати місце проживання іноземного студента шляхом вибору потрібного гуртожитку зі списку доступних;

- замість фіксованого набору довідок, який не можна було змінювати, створений репозиторій з шаблонами довідок. Кожен шаблон можна змінити в будь-який момент часу;

- робота з генератором звітів, що дозволяє створювати звіти про студентів на основі обраних критеріїв.

Доступ до даних про оплати за контрактами неможливий, так як ці дані є конфіденційною інформацією контрактного відділу. Деканат по роботі з іноземними студентами взаємодіє з наступними підрозділами: приймальна комісія для іноземних абітурієнтів, деканати факультетів, адміністрація університету, паспортний відділ, відділ контрактного навчання.

У функціональність підсистеми відділу контрактного навчання були додані:

* Зміни алгоритму нарахування для аспірантів і студентів підготовчого факультету;
* Можливість переведення студентів на наступний рік за наказом;
* Реалізація пені за заборгованість з оплати;
* Можливість переводити студентів, які вийшли, на приклад в академічну відпустку, в лист очікування, де для них не будуть нараховувати нарахування;
* Робота з генератором звітів, що дозволяє створювати звіти про студентів на основі обраних критеріїв.

Функції адміністратора включають всі функції звичайного оператора і, крім того, твір нарахувань на місяць (тільки відділ контрактного навчання), редагування оплат (тільки відділ контрактного навчання) і проведення системних установок системи. Адміністратор має повноправний доступ до всієї інформації, що зберігається в БД.

***2.6 Архітектура системи***

Система «ВКО» складається з клієнтських місць і серверної частини. Для обміну інформацією між ними використовуються мережеві засоби. Між деканатом по роботі з іноземними студентами, паспортним відділом, приймальною комісією для іноземних студентів і відділом контрактного навчання був налаштований VPN-канал. Це дозволило об'єднати всі клієнтські місця в одну локальну мережу.

Сервер володіє функціональністю, яка дозволяє взаємодіяти всім відділам, не заважаючи роботі один одному. Для обміну даними клієнтських місць з серверної частиною використовується технологія RMI (Remote Method Invocation, програмний інтерфейс виклику віддалених методів в мові Java). Дана технологія надає можливість отримувати дані з сервера, тобто з бази даних і зберігати дані в базу даних. Технологія RMI дозволила розділити функціональність системи «ВКО» між усіма клієнтськими місцями.

І клієнтська, і серверна частина розроблена з урахуванням можливості подальшого розширення системи «ВКО».

## 2.7 Використовувані технології

Для розробки системи «ВКО» було вибрано наступне програмне забезпечення: мова програмування JAVA і сервер баз даних SQL Server 2012.

JAVA є одним з найактуальніших мов програмування на сьогоднішній день. Java - об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems.

Програми на Java транслюються в байт-код, що виконується будь віртуальною машиною Java (JVM) - програмою, обробній байтовий код і передавальної інструкції обладнанню як інтерпретатор.

Гідність подібного способу виконання програм - в повній незалежності байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання.

Microsoft SQL Server - система керування базами даних (СКБД), розроблена корпорацією Microsoft. Основний використовуваний мову запитів - Transact-SQL, створений спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO по структурованого мови запитів (SQL) з розширеннями. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства; конкурує з іншими СКБД в цьому сегменті ринку.

Сервер баз даних Microsoft SQL Server в якості мови запитів використовує версію мови SQL, що отримала назву Transact-SQL (скорочено T-SQL). Мова T-SQL є реалізацією SQL-92 (стандарт ISO для мови SQL) з множинними розширеннями. T-SQL дозволяє використовувати додатковий синтаксис для збережених процедур і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим додатком).

При взаємодії з мережею Microsoft SQL Server і Sybase ASE використовують протокол рівня додатка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних). Протокол TDS також був реалізований в проекті FreeTDS з метою забезпечити різним додаткам можливість взаємодії з базами даних Microsoft SQL Server і Sybase.

Для забезпечення доступу до даних Microsoft SQL Server підтримує Open Database Connectivity (ODBC) - інтерфейс взаємодії додатків з СКБД. Компанія Microsoft також випустила сертифікований драйвер JDBC, що дозволяє додаткам під керуванням Java (таким як BEA і IBM WebSphere) з'єднуватися з Microsoft SQL Server 2012.

Для отримання даних з сервера і їх запис в базу даних була використана технологія RMI (Remote Method Invocation) - програмний інтерфейс виклику віддалених методів в мові Java.

Розподілена об'єктна модель, специфікує, яким чином здійснюється виклик віддалених методів, які працюють на інший віртуальній машині Java. При доступі до об'єктів на іншому комп'ютері можливо викликати методи цього об'єкту. Необхідно тільки доставити параметри методу на інший комп'ютер, повідомити об'єкту про необхідність виконання методу, а потім передати назад повертається значення. Механізм RMI дає можливість організувати виконання всіх цих операцій.

У термінах RMI об'єкт, який викликає віддалений метод, називається клієнтським об'єктом, а віддалений об'єкт - серверним об'єктом. Комп'ютери виступають в ролі клієнта і сервера тільки для конкретного виклику. Цілком можливо, що при виконанні наступної операції ці комп'ютери поміняються ролями, тобто сервер попереднього виклику може сам стати клієнтом при зверненні до об'єкта на іншому комп'ютері.

При виклику методу віддаленого об'єкта насправді викликається звичайний метод мови Java, інкапсульований в спеціальному об'єкті-заглушці (stub), який є представником серверного об'єкта. Заглушка знаходиться на клієнтському комп'ютері, а не на сервері. Вона упаковує параметри віддаленого методу в блок байтів. Кожен параметр кодується за допомогою алгоритму, що забезпечує незалежність від апаратури. Процес кодування параметрів називається розгортанням параметрів (parameter marshaling). Основна мета розгортання параметрів - перетворення їх в формат, придатний для передачі параметрів від однієї віртуальної машини до іншої.

Метод, що належить заглушці, створює блок, до якого входять такі елементи:

▪ ідентифікатор віддаленого об'єкта;

▪ опис викликається методу;

▪ розгорнуті параметри.

Потім метод заглушки посилає цю інформацію сервера. Далі об'єкт-одержувач виконує для кожного виклику віддаленого методу наступні дії:

▪ згортання параметрів;

▪ пошук викликаного об'єкта;

▪ виклик заданого методу;

▪ вилучення та розгортання повертається або виключення, згенерованого за цим методом;

▪ передача пакета, що складається з розгорнутих повертаються даних, об'єкту-заглушці на клієнтському комп'ютері.

Клієнтський об'єкт-заглушка згортає повертається значення або виключення, отримане з сервера. Результат згортання стає повертається значенням методу заглушки. Якщо віддалений метод повертає виняток, то об'єкт-заглушка повторить його в середовищі об'єкта-клієнта.

Для виклику віддаленого методу використовується той же синтаксис, що і для звернення до локального методу.

Для автоматизації календарних процесів в проекті була інтегрована бібліотека з відкритим кодом Quartz Job Scheduler.

Quartz - це повнофункціональний інструмент для планування завдань, який може бути інтегрований в додаток будь-якої складності, починаючи з найпростіших локальних додатків і закінчуючи найбільшими системами рівня підприємств. Quartz може використовуватися для планування простих і складних завдань, для управління виконанням десятків, сотень і навіть тисяч завдань. Quartz використовують в своїх продуктах такі компанії як Apache Software Foundation, JBoss, Cisco і багато інших.

Для роботи з завданнями (jobs) в Quartz визначено інтерфейс Job в якому реалізований метод:

public void execute (JobExecutionContext context) throws JobExecutionException

Таким чином, для того, щоб визначити всі дії, які ми хочемо виконувати, нам достатньо оголосити клас, який реалізує цей інтерфейс, помістивши всю логіку роботи в метод execute. А JobExecutionContext - це об'єкт, який містить інформацію, яка була передана завданням, перед тим як воно було запущено на виконання.

 В Quartz завдання відокремлені від часу їх виконання - розклад виконання завдань реалізується за допомогою тригерів - спеціальних об'єктів, які містять інформацію про час початку і періодичності завдання. Це дуже зручно, оскільки дозволяє нам запускати завдання на виконання за допомогою абсолютно різних тригерів.

Quartz пропонує два різних засоби, за допомогою яких можна зберігати пов'язані із завданнями і тригерами дані в пам'яті або в базі даних. Перший засіб, екземпляр класу RAMJobStore, є налаштуванням за замовчуванням. Це найпростіше у використанні сховище завдань, що дає до того ж найбільшу продуктивність, оскільки всі дані зберігаються в пам'яті. Головним недоліком цього методу є недостатня збереженість даних. Оскільки всі дані зберігаються в RAM, вся інформація буде загублена після "падіння" додатка або системи.

Щоб виправити ці недоліки, Quartz пропонує JDBCJobStore. Як випливає з назви, цей спосіб збереження завдань поміщає дані в базу даних через JDBC. Ціною більш надійного зберігання даних є більш низька продуктивність, а також велика складність.

Для створення логів була використана технологія Log4j.

Log4j – це надійна, швидка і гнучка платформа для роботи з логами, яка написана на мові програмування Java. Поширюється під ліцензією Apache Software.

Log4j легко конфігурується через зовнішні файли конфігурації під час виконання. Технологія розглядає процес створення логів з точки зору рівнів пріоритетів і пропонує механізми для прямого запису інформації в різноманітні адресати, такі як база даних, файл, консоль.

Робота з логами є важливою складовою розробки програмного забезпечення. Добре написаний код для створення логів забезпечує швидке налагодження, простоту обслуговування і структуроване зберігання інформації під час виконання програми.

Логи також мають свої недоліки. Вони можу сповільнити роботу програми. Якщо лог занадто багатослівний, то це може викликати scrolling blindness (сліпоту прокрутки). Щоб позбавитися цих проблем, log4j розроблений, щоб бути надійним, швидким і розширюваним.

Оскільки створення журналів логу рідко є основним напрямком додатку, API log4j прагне бути простим для розуміння і використання.

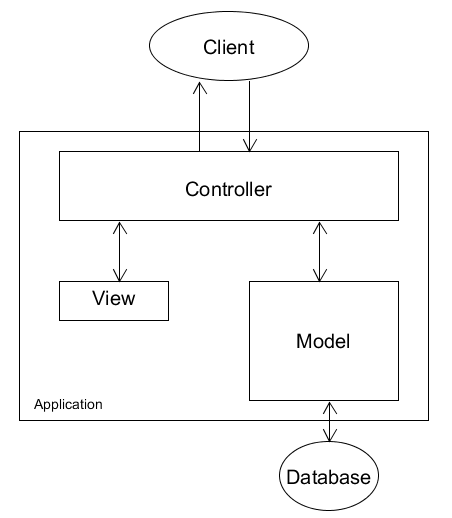
## 2.8 Використані шаблони проектування

При проектуванні системи використовувався шаблон проектування MVC (Model-View-Controller).

Model-View-Controller («Модель-представлення-контролер») – шаблон проектування, за допомогою якого модель даних програми, керуюча логіка та користувацький інтерфейс розділяються на три окремі компоненти. Особливістю такого поділу є те, що будь-яка модифікація одного із компонентів надає мінімальний вплив на інші.

На Мал.3 зображена концепція шаблону Model-View-Controller. Головним аспектом використання даної концепції є те, що виконується відділення бізнес-логіки (моделі) від її візуалізації (представлення).

Шаблон Model-View-Controller визнається багатьма розробниками як шаблон, який дуже вдало підходить для повторного використання коду і шаблон, який дозволяє значно скоротити час, необхідний для розробки додатків з користувацьким інтерфейсом.



Мал.3. Концепція Model-View-Controller

MVC виділяє три основних компоненти, які використовуються в розробці програмного забезпечення:

- Model. Представляє собою логічну структуру даних в додатку та клас, пов'язаний з нею. Ця об'єктна модель не містить будь-якої інформації про користувальницький інтерфейс.

- View. Представляє собою набір класів, що представляють елементи, які призначені для відображення користувацького інтерфейсу (все те, що користувач може бачити на екрані).

- Controller. Класи, що зв'язують модель і представлення. А також використовуються для обміну даних між класами в моделі і представлення.

## 2.9 Розробка діаграми пакетів.

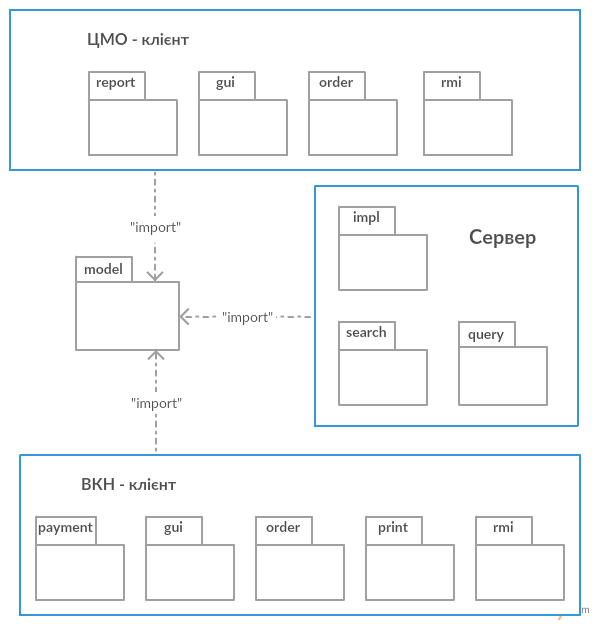
Структура системи складається з великої кількості класів, що пов'язано з великою кількістю різноманітних даних, які необхідно обробляти, відображати користувачеві, витягувати і зберігати в базу даних. Всі розроблені класи були розміщені по пакетах. На наведеній нижче діаграмі (Рис. 7) видно, що деякі класи є однаковими для обох видів клієнтів. Це є причиною перекривається функціональність всіх відділів. Наприклад, складанням звітів та пошуком даних про зареєстрованих в базі даних студентів займаються кілька з розглянутих відділів.

Розглянемо призначення пакетів.

Серверна частина:

* «impl»: забезпечує функціональністю збереження і отримання даних з бази даних; дозволяє клієнтським місцях взаємодіяти з базою даних; містить класи, які відповідають за відправлення даних у вигляді об'єктів на запити про отриманні даних з бази даних.
* «search»: містить класи, які дозволяють здійснювати пошук в базі даних і коректно витягати.
* «query»: містить класи, які дозволяють формувати різноманітні запити до сервера даних для подальшого вилучення інформації.

*Мал. 4*. «*Діаграма пакетів»*



ЦМО-клієнт:

* «report»: класи, що відповідають за створення різноманітних звітів про іноземні абітурієнтів і студентів для відділів центру міжнародної освіти; звіти створюються по шаблонах, які були отримані від співробітників відділів центру міжнародної освіти.
* «gui»: класи, що відповідають за подання.
* «order»: класи, що відповідають за створення наказів про зарахування, переведення, поновлення, відрахування іноземних студентів і абітурієнтів.
* «rmi»: класи, що містять інтерфейс для доступу до реалізації RMI-сервісів ЦМО-клієнта, розташованих на серверної частини системи.

Класи клієнтського місця відділу контрактної освіти були взяті з попередньої версії системи і модернізовані в зв'язку з поставленими завданнями. ОКО-клієнт:

* «payment»: класи, що відповідають за реєстрацію та редагування оплат студентів і абітурієнтів контрактної форми навчання
* «gui»: класи, що відповідають за подання.
* «order»: класи, що відповідають за створення наказів про зарахування, переведення, поновлення, відрахування студентів та абітурієнтів контрактної форми навчання.
* «rmi»: класи, що містять інтерфейс для доступу до реалізації RMI-сервісів ОКО-клієнта, розташованих на серверної частини системи.
* «print»: класи, що відповідають за друк різних документів, довідок і звітів відділу контрактної освіти.

Пакет «model» містить класи, які представляють сутності системи (модель). Всі інші пакети залежать від моделі, так як використовують і імпортують її.

## 2.10 ***Опис нововведень***

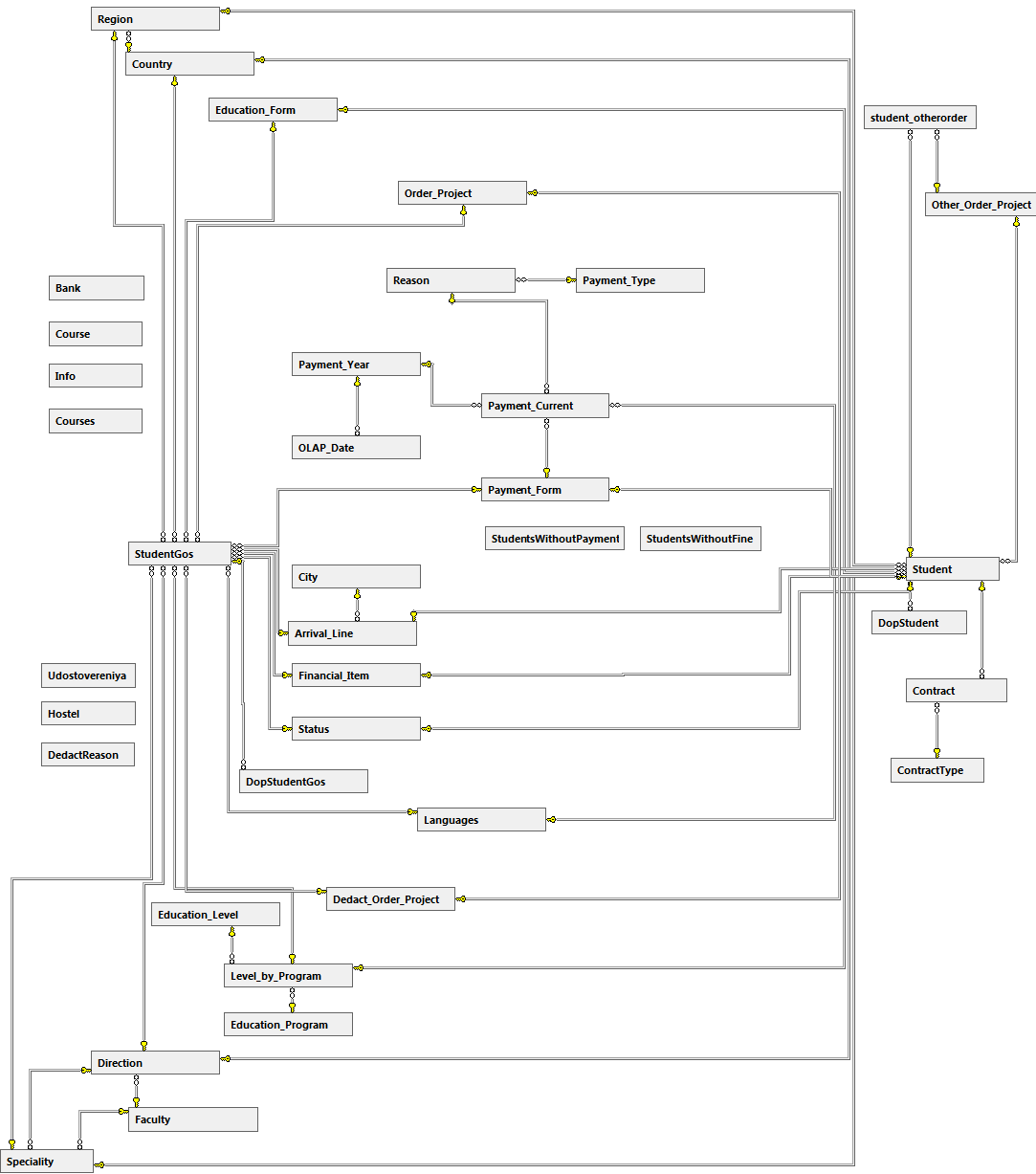
### 2.10.1 Розширення Бази Даних

За основу для проекту була взята оновлена ​​база даних.

 У базу даних були додані наступні таблиці:

* Таблиця **Hostel**, що містить перелік усіх гуртожитків;
* Таблиця **Other\_Order\_Project**, що зберігає інформацію по інших звітів (доданим до цієї версії);
* Таблиця **Udostovereniya**, зберігає інформацію про посвідчення, про тимчасове проживання (для іноземних студентів);
* Таблиця **StudentsWithoutPayment**, містить список відсторонених від навчання студентів (академічну відпустку), яким не повинні йти нарахування за контрактом;
* Таблиця **StudentWithoutFine**, містить список студентів, яких не нараховується пеня за заборгованістю;
* Таблиця **DedactReason**, містить перелік причин на відрахування;
* Таблиця **Instructions**, містить інструкції для планувальника.

Також були внесені зміни до вже існуючих таблиця. Наприклад, в таблицю **DopStudent** були додані нові змінні. Всі нові додаткові дані, що стосуються нострифікації (нострифікація - процедура визнання документів іноземних держав) тепер зберігаються в цій таблиці.



*Мал. 5 Схема бази даних оновленої системи*

### 2.10.2 Розширення віддалених методів (RMI)

Багато методів, взяті з попередньої версії, були змінені відповідно до поставлених завдань з модернізації системи.

  Крім того до вже існуючих було додано ряд нових:

* getDedactReason ()

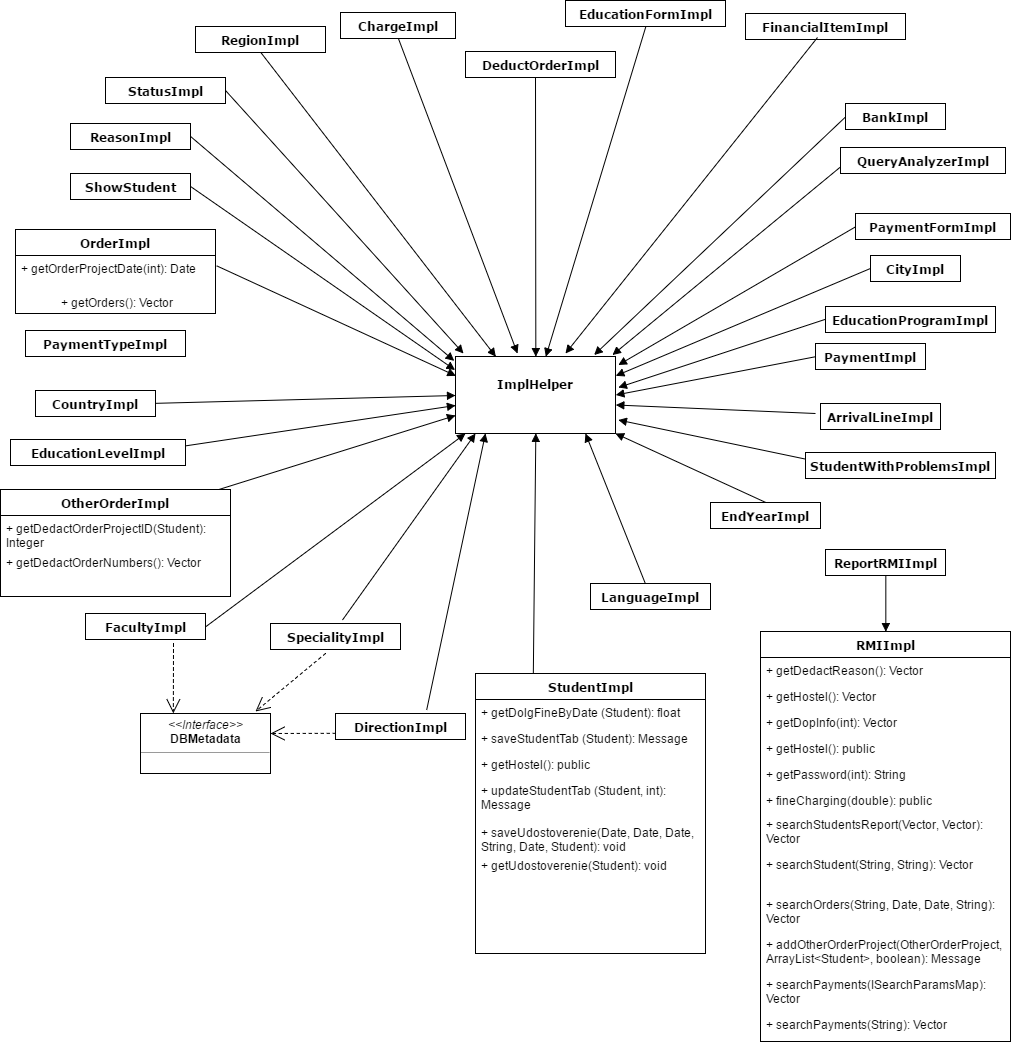
public Vector getDedactReason ()

Метод викликає список причин на відрахування з таблиці DedactReason.

* getHostels ()

public Vector getHostels ()

Метод викликає список гуртожитків з таблиці Hostel.



*Мал. 6. Діаграма класів пакета rmi / impl*

● getDopInfo (int id)

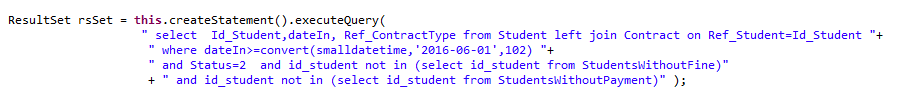
public Vector getDopInfo (int id)

Раніше цей метод надавав інформацію для документів про зарахування (відрахування), а тепер він також дає дані для інших звітів.

● fineCharging (double rate)

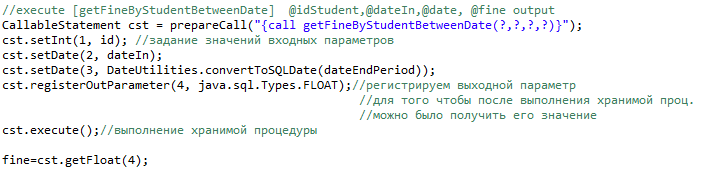
public synchronized Message fineCharging (double rate)

Нарахування пені за минулий місяць відбувається на початку наступного (пов'язано з реєстрацією оплат) тільки для студентів (не аспіранти і не студенти під. фак-ту), зарахованих з 01.06.2016 і у яких немає в таблицях на виключення і в таблиці студентів із припиненими нарахуваннями.



Далі в залежності від того чи був студент зарахований в попередньому місяці, тобто в місяці за який закривається пеня чи ні метод передбачає два варіанти підрахунку. Обидва варіанти реалізовані в процедурах, так як це дозволяє спростити процес зміни логіки підрахунку пені в майбутньому.

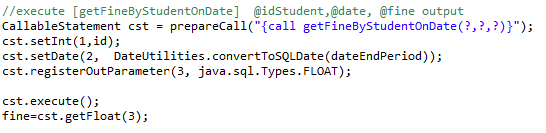
Для студентів зарахованих в попередньому місяці логіка реалізована в збереженій процедурі *getFineByStudentBetweenDate.*



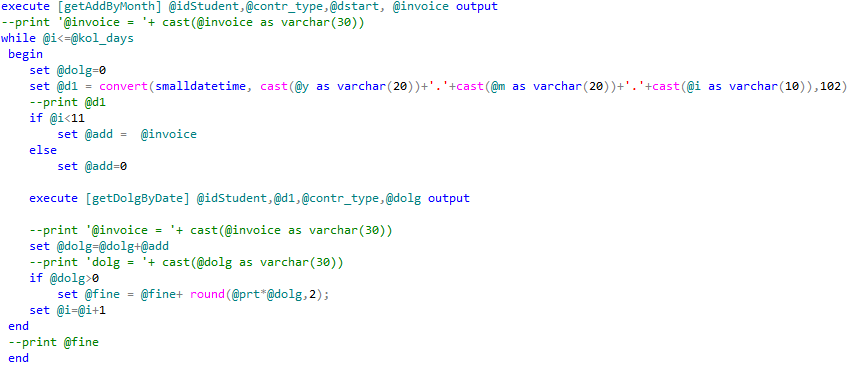
В цьому випадку студентам даються 11 пільгових днів після закінчення, яких тільки починається підрахунок пені (0.05 за кожен прострочений день).



Для студентів зарахованих в попередньому місяці логіка реалізована в збереженій процедурі *getFineByStudentOnDate.*



Перші 10 днів пеня вважається на основі заборгованості без урахування нарахувань за поточний місяць, а починаючи з 11 дня вже з урахуванням.



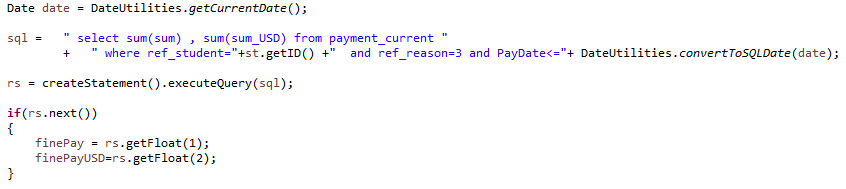
Раніше в базі даних не було відображено дата зарахування, що робило неможливим підрахунок пені. В оновленій версії системи це було враховано, так що тепер кожен студент має крім дати підписання наказу ще й дату зарахування.

● getDolgFineByDate (Student st)

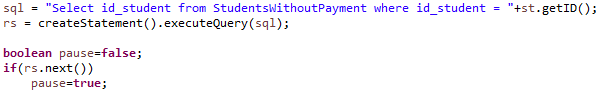
public float getDolgFineByDate (Student st)

Метод отримання боргу пені в картці студента. Також як fineCharging () цей метод працює тільки для тих студентів, які були зараховані після 01.06.2016.

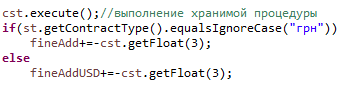
Спочатку вираховується суми до сплати (в гривнях та доларах).



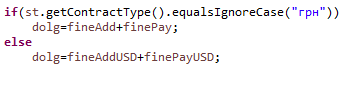
Після для студентів, які не стоять в списку з припиненими нарахуваннями (на паузі) проводять підрахунок пені за допомогою збережених процедур *getFineByStudentOnDate* та *getFineByStudentBeteewnDate* описаних вище.



Отримана пеня помножається на (-1).



У підсумку ми отримуємо результат:



● saveUdostoverenie (Date Date1, Date Date2, Date Date3, String text, Date Date4, Student student)

public void saveUdostoverenie (Date Date1, Date Date2, Date Date3, String text, Date Date4, Student student)

Додає в таблицю Udostoverenie запис.

● getUdostoverenie (Student student)

public Vector getUdostoverenie (Student student)

Вибирає запис з таблиці Udostoverenie.

● getOrderProjectDate (int id)

public synchronized Date getOrderProjectDate (int id);

Метод повертає дату заданого наказу.

● searchStudent (String order, String type)

public Vector searchStudent (String order, String type)

Методу передається номер і тип (зарахування / відрахування / інший) наказу.

На вихід подається перелік студентів.

● searchOrders (String order, Date datestart, Date Dateout, String OrderType)

public Vector searchOrders (String order, Date datestart, Date Dateout, String OrderType)

Метод реалізує пошук наказів по номеру і типу (на зарахування, відрахування та іншому) наказу, в проміжку між датами.

● getDedactOrderNumbers ()

public Vector getDedactOrderNumbers ()

Метод повертає номери наказів з таблиці з переліком відрахованих студентів Dedact\_Order\_Project.

● getDedactOrderProjectID (Student st)

public Integer getDedactOrderProjectID (Student st)

Метод отримання id наказу на відрахування по вводиться номеру студента.

● getOrders ()

public Vector getOrders ()

Метод з отримання списку зареєстрованих наказів на зарахування.

● addDeductOrderProject (DedactOrderProject proj, ArrayList <Student> v, boolean check)

public Message addDeductOrderProject (DedactOrderProject proj, ArrayList <Student> v, boolean check)

Метод додає наказ на відрахування для контрактників і бюджетників.



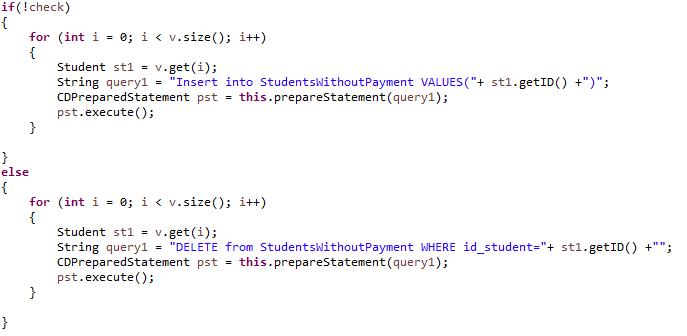
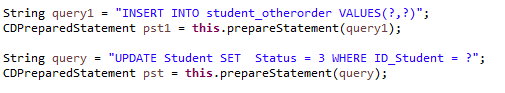
Також метод видаляє записи по нарахуваннях з таблиці Payment\_Current для контрактників.



● addOtherOrderProject (DedactOrderProject proj, ArrayList <Student> v, boolean check)

public Message addOtherOrderProject (DedactOrderProject proj, ArrayList <Student> v, boolean check)

Крім того що цей метод додає інші проекти в ньому також реалізовано додавання і видалення студентів з таблиці із припиненими нарахуваннями по іншим наказом.

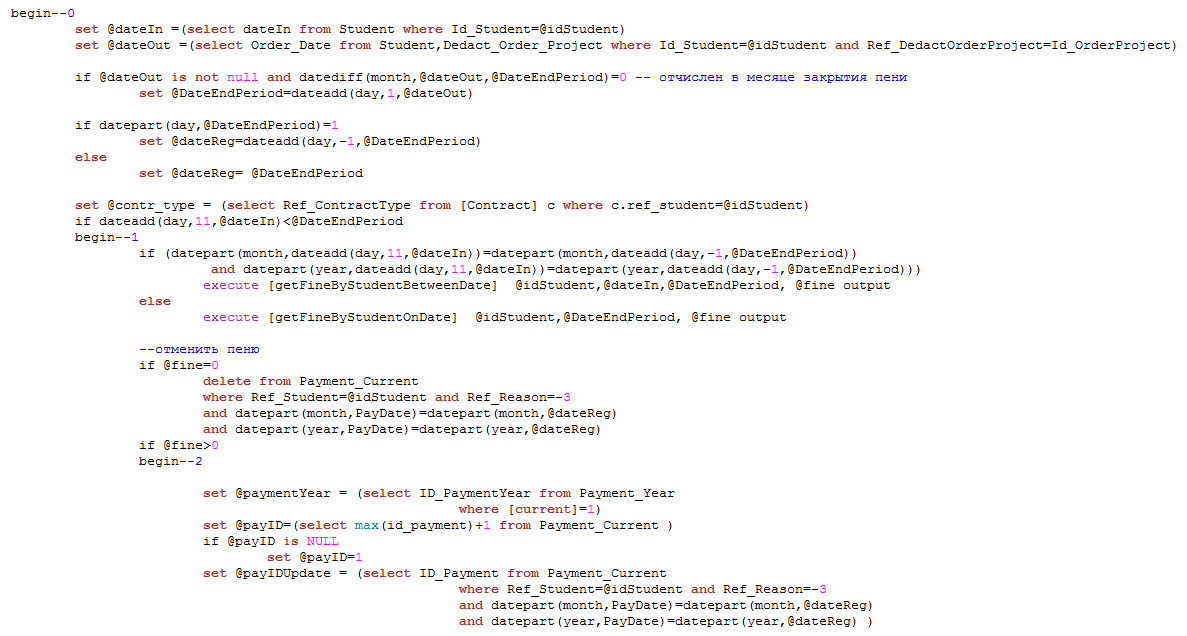
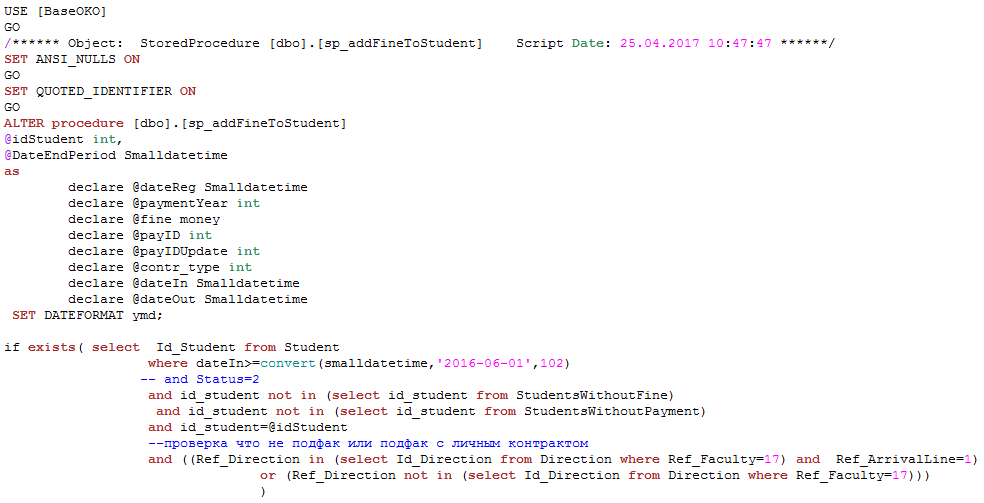


Логіку нарахувань було вирішено винести в окремі процедури, що для спрощення впровадження змін в майбутньому. Були розроблені наступні процедури:

**sp\_addFineToStudent:**

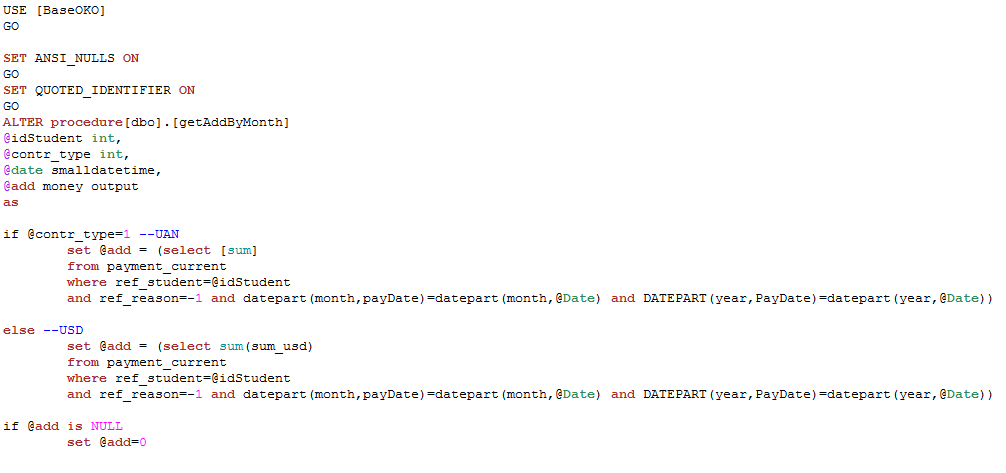
Ця процедура підраховує пеню та зберігає її в таблицю Payment\_current.

В ній також застосовуються процедури *getFineByStudentBetweenDate* та *getFineByStudentOnDate.*



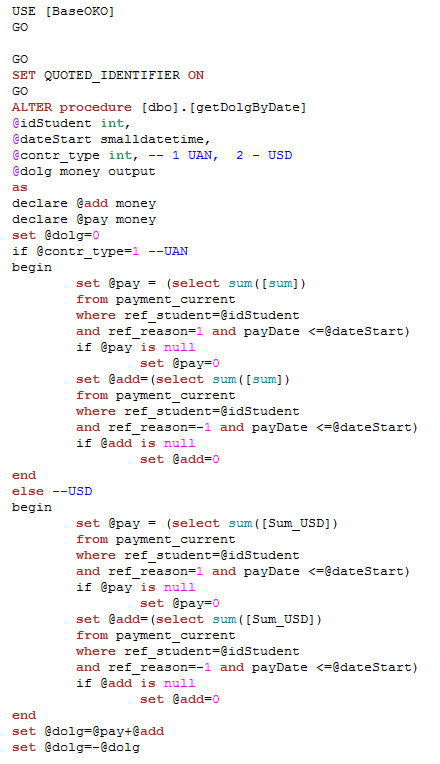
**getAddByMonth:**

Процедура видає суму по нарахування за місяц вказанної дати.



**getDolgByDate:**

В залежності від типу контракту процедура видає суму заборгованності на вказану дату.



# РОЗДІЛ 3. ВИСНОВОК

В результаті виконання даної дипломної роботи, була розроблена архітектура серверної частини інформаційних систем «Відділ контрактної освіти» і  «Центр міжнародної освіти»» Харківського національного університету імені В.Н.Каразіна.

Було визначено актуальність проекту. У розробці проекту було зацікавлене керівництво Харківського національного університету імені В.Н. Каразіна, так як при успішному впровадженні цього проекту підвищилося б якість, зручність і надійність роботи відділів. За участю співробітників всіх вищеназваних відділів була визначена необхідна функціональність і вимоги планованої системи.

Автоматизовані системи «ВКО» та «ЦМО» були оновлені для работи на сучасних платформах. Також все програмне забеспечння використане в розробці є вільно поширюваним, що дозволило уникнути фінансових труднощів з покупками ліцензій.

Проект пройшов етапи збору вимог, планування, проектування і розробки. На підставі успішного впровадження нових функціональностей було прийнято рішення про впровадження системи в роботу. Перед тим як приступити до роботи, працівники задіяних відділів будуть попередньо ознайомлені з новою функціональністю і можливостями розробленої системи.

Такими чином, поставлені цілі і завдання були повністю досягнуті в рамках розробленого проекту.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Еккель Б. Філософія Java (4-е видання). - СПб .: Пітер, 2011.
2. Шилдт Г. Java 8. Повне керівництво. - М.: Видавничий дім "Вільямс",2017.
3. Фаулер М. Архітектура корпоративних програмних додатків. - М., Вільямс, 2004.
4. Гамма Е., Хелм Р., Джонсон Р., Вліссідес Дж. Прийоми об'єктно-орієнтованого проектування. Патерни проектування. - СПб .: Пітер, 2001.
5. Град М. Шаблони проектування в JAVA. Каталог популярних шаблонів проектування, проілюстрованих за допомогою UML. - М .: «Нове знання», 2004.
6. Іцик Бен-Ган. Microsoft SQL Server 2012. Основи T-SQL. - "Ексмо", 2015.
7. Техническая документация по отделу контрактного обучения.
8. Джеймс Р. Грофф, Пол Н. Вайнберг - SQL. Полное руководство
9. <http://www.microsoft.com/sqlserver> - техническая документация по SQL Server 2000.