МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КОЛЕДЖ РАКЕТНОКО-СМІЧНОГО МАШИНОБУДУВАННЯ

ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ім. О. ГОНЧАРА

Предметна комісія програмної інженерії

**КУРСОВИЙ ПРОЕКТ**

з навчальної дисципліни

"ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ"

на тему: «Лінійне програмування. Рішення транспортних задач»

(вказати тему курсового проекту)

Студента \_\_\_IV\_\_ курсу ПЗ-16-1 групи

спеціальності 121 Інженерія

програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_Вайчекаускас С.К. \_\_\_\_\_\_\_\_\_

(прізвище та ініціали студента)

Керівник викладач Н.В Гапоненко

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_ Оцінка ECTS: \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_ \_В.В.Сітарчук\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_ С.С.Ланська

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_ Н.В.Гапоненко

(підпис) (прізвище та ініціали)

Дніпро

2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КОЛЕДЖ РАКЕТНО-КОСМІЧНОГО МАШИНОБУДУВАННЯ

ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ім. О. ГОНЧАРА

Предметна комісія програмної інженерії

|  |  |
| --- | --- |
|  | **ЗАТВЕРДЖУЮ**  Голова предметної комісії ПІ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.С.Ланська  " 0 " nullptr\_\_2019 р. |

**ЗАВДАННЯ**

**на виконання курсового проекту**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| з дисципліни | | | «Об’єктно-орієнтоване програмування» | | | | | | | |
| студенту | Вайчекаускас Станiславу Костянтиновичу | | | | | | | | | |
|  | (прізвище, ім'я та по батькові) | | | | | | | | | |
| Відділення | | Комп'ютерної та програмної інженерії | | | | | | | | |
| Спеціальність | | | | 121 Інженерія програмного забезпечення | | | | | | |
| Курс | IV | | | | | |  | Група (шифр) | ПЗ-16-1 | |
| 1 Тема проекту | | | | | «Експертна система» | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
| 2 Початкові дані: | | | | | | інформація про дисципліни, модулі, компетенції, | | | | |
| коефіцієнт відповідності між компетенціями, питання для перевірки знань. | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
| Розглянуто і ухвалено на засіданні предметної комісії | | | | | | | | | |  |
| Програмної інженерії | | | | | | | | | | |
| Протокол № 6 від 22.01.2018 р. | | | | | | | | | | |

Керівник КП \_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В.Гапоненко\_\_

Завдання до виконання

одержав студент \_\_\_\_\_\_\_\_\_\_\_\_\_ Вайчекаускас С.К.

(підпис) (ініціали та прізвище)

Дата видачі 0 NULL 5780 р.

Термін виконання 0 NULL 5780 р.

ЗМІСТ

[ВСТУП 4](#_Toc21436468)

[1 ПОСТАНОВКА ЗАДАЧІ 5](#_Toc21436469)

[2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ 6](#_Toc21436470)

[2.1 Опис середовища програмування 6](#_Toc21436471)

[2.2 Опис мови програмування 7](#_Toc21436472)

[2.3 Опис транспортної задачі 9](#_Toc21436473)

[2.4 Опис мови стилів CSS 9](#_Toc21436474)

[2.5 Опис основних принципів ООП 10](#_Toc21436475)

[3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ 13](#_Toc21436477)

# ВСТУП

В умовах сучасного інформаційного суспільства все актуальнішою стає потреба використання новітніх технологій у навчанні, як для покращення методів та технології навчання, так і для підвищення його ефективності. Також використання новітніх інформаційних технологій дає можливість набагато збільшити стандартизацію знань та методи покращення сприймання їх студентами навчального курсу, та значно зменшити часові та фінансові затрати у різних галузях.

Одним з шляхів розв’язання цієї проблеми є система безперервної освіти, яка складається з базової і подальшої освіти та передбачає на другому етапі послідовне чергування навчання в системі спеціально створених освітніх закладів з професійною діяльністю.

Завдяки своїй змістовій наповненості й необмеженості в часі безперервна професійна освіта має можливості для виконання важливих функцій, а саме:

* загальноосвітньої, компенсуючої (усунення недоліків у базовій освіті, її доповнення новою інформацією, що з'являється в умовах інформаційно-технологічної революції);
* адаптивної (гнучка професійна підготовка, перепідготовка й підвищення кваліфікації з метою оновлення професійного досвіду, здобуття іншого фаху в умовах постійних змін на виробництві, розвитку теле- та радіокомунікацій, комп'ютерного доступу до інформаційних банків даних тощо);
* економічної (задоволення потреб держави, регіонів, різних галузей промисловості, сільського господарства і сфери послуг у конкурентоспроможних фахівцях, підготовлених до впровадження новітніх технологій, техніки та ін.).

Сьогодні процес складання інтегрованих навчальних планів, заснований на досвіді і інтуїції працівників вищої школи потребує серйозного удосконалення та наукового підґрунтя прийнятих рішень. Процес конструювання індивідуального навчального плану студента або групи студентів може являти собою педагогічну технологію, орієнтовану на реалізацію інформаційних технологій за допомогою експертної системи. В умовах скорочених строків навчання застосування експертних систем забезпечує можливість побудови індивідуального навчального плану, що підвищує ефективність процесів навчання, викладання і самоосвіти.

Програмна оболонка для створення індивідуального навчального плану — це складова експертної системи яка забезпечує формування індивідуальних навчальних планів здобувачів в системі «неперервної освіти» та перелік питань для організації вступного фахового випробування.

Використання експертної системи при складанні інтегрованих індивідуальних навчальних планів має ряд переваг. По-перше, з часом система буде розширюватися і накопичувати відповідності компетенцій як з суміжних спеціальностей, так і з інших галузей знань. По-друге, можна складати робочі навчальні плани напрямів підготовки без повторного залучення експертів з різних предметних областей. По-третє, систему легко можна буде перенавчати відповідно до нових вимог складання інтегрованих планів.

Виходячи з вище написаного, можна зробити висновок, що тема курсового проекту є актуальною, а поставлене завдання — своєчасним.

# 1 ПОСТАНОВКА ЗАДАЧІ

Темою курсового проекту є «Експертна система».

Мета проекту: розробити програмний додаток для побудови індивідуального навчального плану.

Програмний додаток повинен:

1. Визначати, згідно матриці порівнянь, перелік питань, з яких можуть складатися фахові вступні випробування на наступному освітньому рівні за обраною спеціальністю.
2. Визначати індивідуальний інтегрований навчальний план здобувача або групи здобувачів з урахуванням компетентностей, які здобувач успішно отримав на попередньому освітньому рівні та у визначеній кількості кредитів ЄКТС.
3. Для досягнення поставленої мети експертна система повинна використати математичну модель оптимального розподілу кредитів на базі нечіткої логіки.

# 2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ

## 2.1 Опис середовища програмування

Програма курсового проекту написана в середовищі Qt Creator Community Edition.

 Qt Creator — крос-платформовий інструментарій розробки програмного забезпечення (ПЗ) мовою програмування C++. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних систем (ОС), просто компілюючи текст програми для кожної операційної системи без зміни серцевого коду. Містить всі основні класи, які можуть бути потрібні для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу й закінчуючи класами для роботи з мережею, базами даних, OpenGL, SVG і XML. Бібліотека дозволяє керувати потоками, працювати з мережею та забезпечує крос-платформовий доступ до файлів.

Qt Creator має вбудований редактор форм, що дає змогу власноруч прописувати стиль форми можливостями мови CSS. Підтримує CMake, що дає змогу писати крос-платформові додатки. З відомих додатків Qt використовує Google Earth, завдяки зручності використання цієї IDE .

Була обрана за можливість розробляти кроссплатформенні проекти, легке перенесення проекту з однієї ОС у іншу, можливість написання мобільних додатків мовою С++, широкий вибір інструментів для розробки программного забезпечення та зручний інерфейс.

## 2.2 Опис мови програмування

Програма курсового проекту написана мовою програмування С++. Текст програми знаходиться у додатку Д.

C++, яка спочатку проектувалася, як розширення для С, а також на Java та C#, які запозичили у С синтаксис.

Мова програмування C++ найбільш поширена серед розробників програмного забезпечення. Вона є дуже зручною у розробці прикладних програм; драйверів пристроїв; розробка ОС; відео ігор. Реалізацією мови С++ займаються одночасно декілька проектів як безкоштовних, так і комерційних, а саме: GNU, Microsoft і Embarcadero (Borland).

C++ －високорівнева мова програмування загального призначення, поєднує властивості як високорівневих, так і низькорівневих мов програмування. Здатна до підтримки об'єктно-орієнтованого, узагальненого та процедурного програмування. Від свого попередника, мову програмування С, С++ відрізняється тим, що найбільшу увагу при розробці цієї мови було приділено підтримці об'єктно-орієнтованого та узагальненого програмування. Назва «мова програмування C++» походить від мови програмування C, в якому унарний оператор ++ позначає інкремент змінної.

На початку 1980-х років мову програмування С++ розробив співробітник фірми Bell Laboratories – Бьерн Страуструп. Створення мови С++ не планувалось, для власних потреб Бьерн розробив ряд удосконалень мови програмування С. Ранні версії мови С++, називались «Cі з класами» та почали з'являтися з 1980 року. Страуструп додав до мови програмування С можливість роботи з класами та об'єктами, поклавши фундамент нової, заснованої на синтаксисі С, мови програмування. Синтаксис C++ був заснований на синтаксисі C, так як Бьерн Страуструп прагнув зберегти сумісність з мовою C.

Нововведеннями С++ порівняно з С є:

* простий і практичний інтерфейс;
* підтримка об'єктно-орієнтованого програмування через класи;
* підтримка узагальненого програмування через шаблони;
* доповнення до стандартної бібліотеки;
* додаткові типи даних;
* обробка винятків;
* простори імен;
* вбудовані функції;
* перевантаження імен функцій;
* посилання і оператори управління вільно розподіленою пам'яттю.

Переваги С++:

* 1. Швидкість роботи програм на С++ майже не змінилась порівняно із С, хоча розробники отримали нові можливості і засоби.
  2. Мова С++ є кросплатформенною.
  3. Можливість роботи на низькому рівні з пам'яттю, адресами, портами.
  4. Можливість створення узагальнених алгоритмів для різних типів даних, їхня спеціалізація і обчислення на етапі компіляції з використанням шаблонів.
  5. Підтримуються різні стилі та технології програмування, включаючи традиційне директивне програмування, ООП, узагальнене програмування, мета програмування (шаблони, макроси).

Недоліки С++:

* погана підтримка модульності. Підключення інтерфейсу зовнішнього модуля через процесорну вставку заголовного файлу (#include) серйозно уповільнює компіляцію при підключенні великої кількості модулів. Для усунення цього недоліку багато компіляторів реалізують механізм компіляції заголовних файлів;
* недостача інформації про типи даних під час компіляції;
* мова C++ є складною для вивчення і для компіляції;
* деякі перетворення типів не інтуїтивні. Зокрема, операція над беззнаковим і знаковим числами видає беззнаковий результат.

## 2.3 Опис СУБД MySQL

MySQL - це система управління базами даних.

База даних являє собою структуровану сукупність даних. Ці дані можуть бути будь-якими — від простого списку майбутніх покупок до переліку експонатів картинної галереї або величезної кількості інформації в корпоративній мережі. Для запису, вибірки й обробки даних, що зберігаються в комп'ютерній базі даних, необхідна система управління базою даних, якою і є ПО MySQL. Оскільки комп'ютери чудово справляються з обробкою великих обсягів даних, управління базами даних відіграє центральну роль в обчисленнях. Реалізовано таке управління може бути по-різному - як у вигляді окремих утиліт, так і у вигляді коду, що входить до складу інших додатків.

MySQL - це система управління реляційними базами даних.

В реляційній базі даних дані зберігаються не всі скопом, а в окремих таблицях, завдяки чому досягається виграш в швидкості і гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мову структурованих запитів й одну з найвикористовуваних мов для доступу до баз даних.

Програмне забезпечення MySQL - це ПЗ з відкритим кодом.

ПЗ з відкритим кодом означає, що застосовувати і модифікувати його може будь-хто. Таке ПЗ можна отримувати по Internet і використовувати безкоштовно. При цьому кожен користувач може вивчити вихідний код і змінити його відповідно до своїх потреб. Використання програмного забезпечення MySQL регламентується ліцензією GPL (GNU General Public License), в якій зазначено, що можна і чого не можна робити з цим програмним забезпеченням в різних ситуаціях. Якщо робота у рамках GPL вас не влаштовує або планується вбудовування MySQL-коду в комерційний додаток, є можливість купити комерційну ліцензовану версію у компанії MySQL AB. See section 1.6.3 ліцензії на ПЗ MySQL.

MySQL є дуже швидким, надійним і легким у використанні. MySQL має також ряд зручних можливостей, розроблених в тісному контакті з користувачами. Спочатку сервер MySQL розроблявся для управління великими базами даних з метою забезпечити більш високу швидкість роботи в порівнянні з існуючими на той момент аналогами. І ось вже протягом декількох років даний сервер успішно використовується в умовах промислової експлуатації з високими вимогами. Незважаючи на те що MySQL постійно вдосконалюється, він вже сьогодні забезпечує широкий спектр корисних функцій. Завдяки своїй доступності, швидкості і безпеки MySQL дуже добре підходить для доступу до баз даних по Internet.

Технічні можливості СУБД MySQL

MySQL є системою клієнт-сервер, яка містить багатопоточний SQL-сервер, що забезпечує підтримку різних обчислювальних машин баз даних, а також кілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API). Також поставляється у вигляді многопоточної бібліотеки, яку можна підключити до користувача додатком і отримати компактний, швидкий і легкий в управлінні продукт.

## 2.4 Опис мови стилів CSS

CSS — спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

CSS використовується авторами та відвідувачами веб-сторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерах та ін.).

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного CSS.

## 2.5 Опис основних принципів ООП

Об'єктно-орієнтоване програмування або ООП (object - oriented programming) - методологія програмування, заснована на представленні програми у вигляді сукупності об'єктів кожен з яких являється реалізацією визначеного типу, що використовує механізм пересилки повідомлень і класи організовані в ієрархію наслідування.

Центральний елемент ООП - абстракція. Дані за допомогою абстракції перетворяться в об'єкти, а послідовність обробки цих даних перетворюється на набір повідомлень, що передаються між цими об'єктами. Кожен з об'єктів має свою власну унікальну поведінку. До об'єктів можна звертатися як до окремих сутностей, які реагують на повідомлення, що наказують їм виконати якісь дії.

ООП характеризується наступними принципами (по Алану Кею) :

1. Усе являється об'єктом;
2. Обчислення здійснюються шляхом взаємодії (обміну даними) між об'єктами, при якому один об'єкт вимагає, щоб інший об'єкт виконав деяку дію; об'єкти взаємодіють, посилаючи і отримуючи повідомлення; повідомлення - це запит на виконання дії, доповнений набором аргументів, які можуть знадобитися при виконанні дії;
3. Кожен об'єкт має незалежну пам'ять яка складається з інших об'єктів
4. Кожен об'єкт є представником класуякий виражає загальні властивості об'єктів даного типу
5. У класі задається функціональність (поведінка об'єкту); тим самим усе об'єкти, які являються екземплярами одного класу, можуть виконувати одні і ті ж дії;
6. Класи організовані в єдину деревовидну структуру із загальним коренем, звану ієрархією спадкоємства; пам'ять і поведінка, пов'язана з екземплярами певного класу, автоматично доступні будь-якому класу, розташованому нижче в ієрархічному дереві.

Абстрагування (abstraction) — метод рішення задачі, при якому об'єкти різного роду об'єднуються загальним поняттям (концепцією), а потім згруповані суті розглядаються як елементи єдиної категорії. Абстрагування дозволяє відокремити логічний сенс фрагмента програми від проблеми його реалізації, розділивши зовнішній опис (інтерфейс) об'єкту і його внутрішню організацію (реалізацію).

Інкапсуляція (encapsulation) — техніка, при якій несуттєва з точки зору інтерфейсу об'єкту інформація ховається усередині нього.

Спадкоємство (inheritance) — властивість об'єктів, за допомогою якої екземпляри класу дістають доступ до даних і методів класів-предків без їх повторного визначення. Спадкоємство дозволяє різним типам даних спільно використовувати один і той же код, призводячи до зменшення його розміру і підвищення функціональності.

Поліморфізм (polymorphism) —- властивість, що дозволяє використовувати один і той же інтерфейс для різних дій; поліморфній змінній, наприклад, може відповідати декілька різних методів.

Поліморфізм перекроює загальний код, що реалізовує деякий інтерфейс, так, щоб задовольнити конкретним особливостям окремих типів даних. Клас (class) - безліч об'єктів, пов'язаних спільністю структури і поведінка; абстрактний опис даних і поведінки (методів) для сукупності схожих об'єктів, представники якої називаються екземплярами класу.

Об'єкт (object) — конкретна реалізація класу, що володіє характеристиками стану, поведінки і індивідуальності, синонім екземпляра.

# 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

## 3.1 Опис структури бази даних

Програма має може мати декілька баз даних. Мінімальна кількість ­– дві.

Перша — головна, яка зберігає інформацію про профілі користувачів, їх ролі, список доступних проектів бази знань та їх опис. ER-діаграми створених баз даних можна побачити на рисунках 3.1.1 та 3.1.2.

**databases**

namedb PK

info

**users**

name PK

role

Рисунок 3.1 — ER-діаграма головної бази даних

name\_discipline

control\_discipline

**discipline**

id\_discipline PK

name\_discipline\_2

control\_discipline\_2

**discipline\_2**

id\_discipline\_2 PK

name\_module

hours\_module

**module**

id\_module PK

id\_discipline FK

name\_module\_2

hours\_module\_2

**module\_2**

id\_module\_2 PK

id\_discipline\_2 FK

info\_compet

type\_compet FK

**competence**

id\_compet PK

**users**

name PK

role

**mod\_comp**

id\_module FK

id\_compet FK

**mod\_comp\_2**

id\_module FK

id\_compet FK

info\_compet\_2

type\_compet FK

**competence\_2**

id\_compet\_2 PK

id\_compet FK

id\_compet\_2 FK

**conformity**

id\_coeff PK

value\_coeff

id\_compet\_2 FK

question

**conformity**

id\_question PK

answer\_a

answer\_c

answer\_d

answer\_b

answer\_true

Рисунок 3.1 — ER-діаграма головної бази даних