

Затверджено

482.362.6050102-02 81 09-1 ЛЗ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

Інститут фізико-технічних та комп'ютерних наук  
Кафедра комп'ютерних систем та мереж

**ВАРІАНТ 9**

**482.362.6050102-02 81 09-1**

(Пояснювальна записка)

Сторінок 10

2015

## **АНОТАЦІЯ**

Документ «Пояснювальна записка» включає основні розділи, а саме: призначення та область застосування розробки, технічні характеристики, очікувані техніко – економічні показники, охорону праці та джерела літератури, які використовувалися при розробці. Документ містить відомості про постановку задачі на розробку програми та опис методів додавання і видалення об'єктів класу, а також виведення інформації про них на екран та у файл.

Наведено опис алгоритму й функціонування програми, вхідні та вихідні дані програми.

**ЗМІСТ**

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1. ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ .....	5
2. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ.....	5
2.1. Середовище об'єктно-орієнтовного програмування .....	5
2.1.1. Основні принципи ООП.....	6
2.1.2 Класи, об'єкти і методи в Delphi.....	8
2.2. Постановка задачі на розробку програми .....	8
3. Опис організації вхідних та вихідних даних .....	9
4. Опис вибору технічних і програмних засобів .....	9

482.362.6050102-02 81 09-1 ЛЗ

**ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

ООП – Об’єктно-орієнтоване програмування

## **ВСТУП**

Розроблена програма призначена для опрацювання даних, введених користувачем, використовуючи основні принципи об'єктно-орієнтованого програмування.

Програма розроблена в середовищі розробки Borland Delphi 7.0 та працює під управлінням операційних систем Windows. Вивід даних програма здійснює на стандартний пристрій виводу – монітор. Інтерфейс користувача (меню, діалогові вікна, кнопки) дозволяє швидко освоїти принципи ООП, функціонування його складових та застосування при вирішенні різноманітних задач.

## **1. ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ**

При роботі з базами даних (БД) необхідно забезпечити збереження всієї інформації та розробити систему її опрацювання. Оскільки зазвичай БД містять значну кількість незалежних даних, до яких має бути відкритий доступ, то зручно користуватися динамічними масивами об'єктів певного класу. Тому даний проект призначений для поглибленого дослідження принципів ООП, особливостей роботи з класами, їх властивостями та методами обробки даних, зокрема для вирішення вище зазначеної задачі.

Результати роботи програми можуть використовуватися у навчальних закладах для упорядкування особистих даних про студентів(учнів) та успішності їх навчальної діяльності.

## **2. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ**

### **2.1. Середовище об'єктно-орієнтовного програмування**

Історично склалося так, що програмування виникло і розвивалося як процедурне програмування, яке припускає, що основою програми є алгоритм, процедура обробки даних.

Об'єктно-орієнтоване програмування (ООП) - це методика розробки програм, в основі якої лежить поняття об'єкту, як деякої структури, що описує об'єкт реального світу, його поведінку. Завдання, що вирішується з використанням методики ООП, описується в термінах об'єктів і операцій над ними, а програма при такому підході є набором об'єктів і зв'язків між ними.

Строго кажучи, для того, щоб розробляти додатки в Delphi на базі тих, що надаються середовищем розробки компонентів, знання концепції ООП не є необхідним. Проте для глибшого розуміння того, як програма взаємодіє з компонентами, що і чому Delphi додає в текст програми, матеріал даного розділу вельми корисний.

### **2.1.1. Основні принципи ООП**

ООП – це стиль програмування, що з'явився в 80 роках 20 століття. На відміну від процедурних мов, де дані і інструкції з їх обробки існують окремо, в об'єктно-орієнтованому програмуванні ця інформація об'єднується в єдину сутність.

У об'єктно-програмного програмування є свої постулати. Принципи ООП – це його основні ідеї. Виділяють три найголовніші з них: успадкування, поліморфізм і інкапсуляція. Нижче кожен буде розглянуто більш докладно. Основи програмування на мовах ООП полягають у використанні об'єктів і класів. При переході від процедурного стилю написання вихідного коду до об'єктно-орієнтованого нерідко виникають складнощі, проте більшість розробників знаходять в ООП безліч плюсів.

Інкапсуляція – це використання об'єднання даних та інструкцій щодо їх обробки в єдину сутність – клас. Під час написання програм на одній з мов ООП відбувається розмежування між інформацією всередині сутності і зовні. Таким чином досягається забезпечення безпеки даних і методів їх реалізації від зовнішніх впливів, наприклад, з боку інших класів, що не відносяться до

цього об'єкта. Усередині сутності дані успішно взаємодіють один з одним, але надійно захищені від несанкціонованого доступу ззовні.

Другий принцип ООП – спадкування – це можливість одного класу використовувати методи іншого без повторення їх фактичної реалізації. Спадкування дозволяє позбутися від надмірності вихідного коду.

Ще один принцип ООП – поліморфізм. Його використання означає, що для маніпуляції з об'єктами різного ступеня складності можна створити один інтерфейс, який буде по-різному реагувати на події і одночасно правильно реалізовувати поставлені завдання.

Принципи ООП використовуються в таких найбільш популярних мовах програмування, як C++ і Java, на яких розроблена значна частина програм та програм. Є й менш використовувані мови ООП – це Delphi, Object Pascal, Ruby, Python і багато інших.

Незважаючи на в основному позитивні висловлювання у бік даної методології, нерідко принципи ООП піддаються і критиці. Як і у процедурного програмування, у ООП є свої недоліки.

По-перше, складність переходу. Щоб зрозуміти принципи ООП, буде потрібно досить багато часу, тим більше людям, впритул працює тільки з процедурними мовами програмування. По-друге, недоліком є більш складна документація, так як потрібно не тільки описувати класи та об'єкти, а й конкретні випадки їх реалізації. По-третє, зайва універсальність методів може призвести до того, що вихідний код і розроблювані програми будуть перевантажені незатребуваними в даному конкретному випадку функціями і можливостями. Крім того, відзначають неефективність з точки зору розподілу пам'яті. Однак незалежно від думки оточуючих число програмістів ООП постійно зростає, а самі мови стрімко розвиваються.

### **2.1.2. Класи, об'єкти і методи в Delphi**

Класична мова Pascal дозволяє програмістові визначати свої власні складні типи даних - записи (records). Object Pascal, підтримуючи концепцію об'єктно-орієнтованого програмування, дає можливість визначати *класи*. Клас - це складна структура, що включає крім опису даних опис процедур і функцій, які можуть бути виконані над представником класу - об'єктом.

Методи класу (процедури і функції, оголошення яких включене в опис класу) виконують дії над об'єктами класу. Щоб метод був виконаний, треба вказати ім'я об'єкту і ім'я методу, відокремивши одне ім'я від іншого крапкою. Наприклад, інструкція

professor.Show;

викликає застосування методу show об'єкту professor. Фактично інструкція застосування методу до об'єкту - це специфічний спосіб запису інструкції виклику процедури.

У програмі методи класу визначаються точно так, як і звичайні процедури і функції, за винятком того, що ім'я процедури або функції, що є методом, складається з двох частин: імені класу, до якого належить метод, і імені методу. Ім'я класу від імені методу відділяється крапкою.

### **2.2. Постановка задачі на розробку програми**

Основна мета курсового проекту опрацювання даних, введених користувачем, використовуючи основні принципи об'єктно-орієнтованого програмування. Доступ користувача до всіх даних на будь-якому етапі роботи та зручне керування, редагування та збереження їх є першочерговим призначенням даної програми.

Основна робота програми полягає в створенні нової інформаційної структури даних (класу) та її опрацювання за допомогою вивчених алгоритмів та принципів об'єктно-орієнтованого програмування.



### **3. Опис організації вхідних та вихідних даних**

Вхідними даними вважається інформація введена користувачем з метою майбутньої роботи з цими даними.

До вихідних даних програми відносяться файл спеціального формату для збереження даних для їх повторного використання, а також виведені данні у вигляді таблиці.

### **4. Опис вибору технічних і програмних засобів**

Розроблений програмний продукт орієнтований на роботу в ОС Windows XP/7/8/8.1, тому для коректної роботи програми необхідне стабільне функціонування ОС. Під час виконання, програма не звертається до інших прикладних програмних продуктів, таких як Microsoft Office та ін.

Вимоги до об'єму оперативної пам'яті – мінімум 64 Мб зумовлені вимогами операційної системи та необхідних обчислень.

Вимоги до об'єму жорсткого диску – мінімум 2 Гб зумовлені вимогами операційної системи та необхідного програмного забезпечення.

Для усунення виявлених помилок та створення завантажувального файлу необхідний встановлений пакет інструментальних засобів Embarcadero Delphi XE.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Батищев Д. И. Генетические алгоритмы решения экстремальных задач. – Воронеж: ВГТУ, 1995. – 231 с. Скобцов Ю.А., Скобцов В.Ю. Логическое моделирование и тестирование цифровых устройств. – Донецк: ИПММ НАН Украины, ДонНТУ, 2005. – 436с.
2. Goldberg D.E. Genetic Algorithms in Search, Optimization & Machine Learning. –Addison-Wesley Publishing Company, Inc., 1989. P.150.
3. Люгер Д., Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е изд. – М.: Вильямс, 2003. – 864 с.