

Міністерство освіти і науки України
Національний університет водного господарства та природокористування
Навчально-науковий інститут автоматики, кібернетики та обчислювальної техніки

Кафедра прикладної математики

«ЗАТВЕРДЖУЮ»

Проректор з науково-педагогічної,
методичної та виховної роботи

_____ О. А. Лагоднюк

“ ___ ” _____ 2019 р.

04-01-39

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
Program of the Discipline

“ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ”

“OBJECT-ORIENTED PROGRAMMING”

Спеціальність 113 «Прикладна математика»
Specialty – 113 «Applied Mathematics»

Рівне – 2019 рік

Робоча програма «Об'єктно-орієнтоване програмування» для студентів які навчаються за спеціальністю 113 «Прикладна математика». – Рівне: НУВГП, 2019. – 15 с.

Розробник:

Жуковський Віктор Володимирович, старший викладач кафедри прикладної математики.

Робоча програма затверджена на засіданні кафедри прикладної математики

Протокол від “28 ” серпня 2019 року № 16

Завідувач кафедри прикладної математики _____ (П. М. Мартинюк)

Схвалено науково-методичною комісією за спеціальністю 113 «Прикладна математика»

Протокол від “ 30 ” серпня 2019 року № 8

Голова науково-методичної комісії _____ (П. М. Мартинюк)

© Жуковський В. В., 2019 рік

© НУВГП, 2019 рік

Вступ

«Об'єктно-орієнтоване програмування» відноситься до циклу дисциплін професійної підготовки бакалаврів спеціальності 113 «Прикладна математика». Програма дисципліни «Об'єктно-орієнтоване програмування» складена відповідно до освітньо-професійної програми спеціальності 113 «Прикладна математика» та стандарту вищої освіти за вказаною спеціальністю для першого (бакалаврського) рівня вищої освіти (затверджений наказом МОН України №1242 від 13.11.2018). Дисципліни, що передують вивченню «Програмування», «Операційні системи».

Дисципліна «Об'єктно-орієнтоване програмування» є базовою для засвоєння інших курсів та спецкурсів з програмування, окремих розділів теорії алгоритмів та математичної логіки.

Анотація

Головна мета курсу – оволодіння технікою об'єктно-орієнтованого програмування на основі мов програмування C# та C++. Увага в курсі об'єктно-орієнтованого програмування концентрується не на складних алгоритмах, а на побудові обґрунтованих ієрархій, складанні абстрактних інтерфейсів та універсальних сигнатур методів. Ідеал об'єктно-орієнтованого програмування – це програми, що придатні до налагодження та функціонування за нових умов лише завдяки дописуванню, але не переписуванню коду.

Дисципліна знайомить студентів із сучасними методами та принципами розробки програмного забезпечення ПК: операційною системою Microsoft Windows, інтегрованим середовищем розробки Visual Studio .NET, об'єктно-орієнтованим програмуванням та мовою C#. Вивчення курсу супроводжується значною кількістю лабораторних робіт, що дозволяє студентам отримати стійкі практичні навички по застосуванню програмних засобів, що вивчаються.

Ключові слова: програмування, клас, об'єкт, Microsoft Visual Studio C#, інкапсуляція, наслідування, поліморфізм, патерни.

Abstract

The main purpose of the course is to master the technology of object-oriented programming based on the programming languages C# and C++. Attention in the course of object-oriented programming is focused on the construction of reasonable hierarchies, the compilation of abstract interfaces and universal signature of methods. Ideal object-oriented programming is a program that is suitable for debugging and functioning under the new conditions only by adding, but not rewriting the code.

The discipline introduces students to the modern methods and principles of PC software development: Microsoft Windows operating system, Visual Studio .NET Integrated Development Environment, object-oriented programming, and C#. The course study is accompanied by a significant amount of laboratory work, which allows students to obtain sustainable practical skills in applying study tools.

Keywords: programming, class, object, Microsoft Visual Studio C#, encapsulation, imitation, polymorphism, patterns of programming.

СТРУКТУРА ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

“Об’єктно-орієнтоване програмування”

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Таблиця 1.1

Найменування показників	Напрямок, спеціальність, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
1	2	3	
Кількість кредитів, відповідних ECTS – 4	Галузь знань: 11 „Математика та статистика”	За вибором ВНЗ. Дисципліни професійної та практичної підготовки	
Модулів – 1	Спеціальність: 113 „Прикладна математика”	Рік підготовки:	
Змістових модулів – 3		2-й	-
		Семестр:	
Загальна кількість годин – 120;		4-ий	-
		Лекцій:	
		24 год.	-
Тижневих годин для денної форми навчання: 12 Ауд.: 4 СРС: 6	Рівень вищої освіти: бакалавр	Лабораторних робіт:	
		24 год.	-
		Самостійна робота:	
		72 год.	-
		Вид контролю: залік	

Примітка. Співвідношення кількості годин аудиторних занять та самостійної і індивідуальної роботи становить 40% до 60%.

2. Мета навчальної дисципліни

У основі створення переважної більшості сучасних програмних продуктів лежить об'єктно-орієнтована методологія. Курс «Об'єктно-орієнтоване програмування» покликаний познайомити студентів із основами ООП, дати поняття про його базові концепції (інкапсуляція, поліморфізм, наслідування), навчити використовувати мови програмування, що підтримують методологію ООП (C++, C#), познайомити зі стандартними алгоритмами та шаблонами (STL). Також у цьому курсі студенти знайомляться із основами UML (Unified Modelling Language), що є стандартним засобом при проектуванні, розробці та тестуванні складних систем та процесів. І, насамкінець, студенти повинні оволодіти методологією програмування із використанням шаблонів проектування (design patterns) та їх класифікацію.

Метою навчальної дисципліни "Об'єктно-орієнтоване програмування" є вивчення та засвоєння основ об'єктно-орієнтованого візуально-подійного програмування з подальшим застосуванням студентами отриманих знань в своїй роботі та навчанні.

Предметом вивчення дисципліни є принципи об'єктно-орієнтованого програмування, а також методи їх використання при розробці додатків.

Завданням вивчення дисципліни є навчити студентів програмувати на мові C#/C++ з використанням об'єктно-орієнтованого підходу.

Після вивчення даного курсу студент повинен **знати**:

- синтаксис, основні елементи та конструкції мови C#;
- об'єктно-орієнтовані методи та підходи до розробки програм.

Студент повинен **вміти**:

- проектувати ієрархію класів для розв'язання поставленої задачі;
- використовувати дизайн-патерни та шаблони проектування;
- створювати програми використовуючи інструментарій Microsoft Visual Studio C++/C#.

3. Програма навчальної дисципліни

Вступ. Предмет „Об'єктно-орієнтоване програмування”, його роль у формуванні світогляду студента, практичне значення та зв'язок з іншими дисциплінами. Огляд літератури.

Змістовий модуль 1. Принципи об'єктно-орієнтованого програмування

Тема 1. Мова програмування C#

Історія створення мови C#. Мова C. Мова C++. Мова Java. Мова C#. Платформа .NET Framework. Принцип роботи загального середовища виконання мов. Керований та некерований код. Ключові слова. Ідентифікатори.

Тема 2. Середовище Microsoft Visual Studio

Використання Visual Studio IDE. Створення проекту. Компіляція з командної стрічки та з середовища розробки. Відлагодження програми.

Тема 3. Класи

Декомпозиція. Абстракція. Ієрархія. Зміст проектування. Об'єктно-орієнтоване програмування, проектування та аналіз. Клас. Створення і використання об'єктів. Інкапсуляція. Наслідування. Поліморфізм. Приклад коду на мові C# та C++.

Тема 4. Конструктори та деструктори

Конструктор. Деструктор. Inline функції. Оператор прив'язки. Область видимості (private, public, protected).

Тема 5. Віртуальні функції

Відношення між класами. Асоціація. Агрегація. Використання. Інстанціювання. Множинне наслідування. Поліморфізм. Віртуальні функції. Механізм раннього зв'язування. Зауваження щодо віртуальних функцій (конструктори, деструктори).

Тема 6. Абстрактні класи

Абстрактні класи. Інтерфейси. Порівняння інтерфейсу та абстрактного класу. Перевантаження операцій. Обмеження на перевантаження операцій.

Тема 7. Стрічки

Створення стрічок. Робота зі стрічками. Стрічкові константи. Клас object. Приведення до об'єктного типу та відновлення значення. Властивості. Класи потоків.

Тема 8. Робота з файлами

Юнікод кодування. Клас FileStream. Класи BinaryReader та BinaryWriter. Класи StringReader та StringWriter.

Тема 9. Організація графічного інтерфесу

Графічний інтерфейс користувача. Windows форми. Створення проекту з формою в Microsoft Visual Studio. Елементи керування. Взаємодія форм. Модальні та немодальні форми. Передача інформації між формами.

Тема 10. Рисування на формі

Клас Graphics. Клас Pen. Клас Brush. Приклад коду для рисування павутина Безьє.

Тема 11. Регулярні вирази

Простір імен RegularExpression. Регулярний вираз. Символи, що використовуються в регулярних виразах. Клас Regex. Клас Match. Клас MatchCollection. Приклад коду.

Змістовий модуль 2. Патерни проектування

Тема 12. Патерни проектування

Основні елементи патерну. Класифікація патернів. Породжуючі патерни. Структурні патерни. Патерни поведінки. Патерни декоратор (Decorator), фасад (Facade), медіатор (Mediator), відвідувач (Visitor).

Тема 13. Патерн абстрактна фабрика (Abstract Factory)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Тема 13. Патерн будівельник (Builder)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Тема 14. Патерн прототип (Prototype)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Тема 15. Патерн одинак (Singleton)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Тема 16. Патерн адаптер (Adapter)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Тема 17. Патерн декоратор (Decorator)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Тема 18. Патерн проксі (Proxy)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Тема 19. Патерн ітератор (Iterator)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Тема 20. Патерн спостерігач (Observer)

Ситуаційна схема. Інтерфейс патерну. Приклад застосування. UML діаграма.

Змістовий модуль 3. Фреймворк Qt

Тема 21. Фреймворк Qt

Опис фреймворку Qt. Встановлення та налаштування. Створення першого додатку в середовищі Qt Designer.

Тема 21. Слоти та сигнали

Механізм слотів та сигналів Qt. Приклад використання. Метаоб'єктна система.

Тема 22. Бібліотека STL

Призначення. Основні можливості. Сфера застосування. Порівняння з класами Qt.

Тема 23. Динамічні масиви

Синтаксис описання динамічних масивів в C++. Відмінність від масивів вказівників. Виділення пам'яті під динамічні масиви. Очищення пам'яті

4. Структура навчальної дисципліни

Таблиця 4.1

Змістові модулі (теми)	Обсяг, годин			
	Денна форма			
	Всього	Лекцій	Лаб. робіт	Сам. робота
1	2	3	4	5
Змістовий модуль 1. Принципи об'єктно-орієнтованого програмування				
Тема 1. Мова програмування C#	3	2	2	1
Тема 2. Середовище Microsoft Visual Studio	5	1	2	2
Тема 3. Класи	9	1	2	6
Тема 4. Конструктори та деструктори	4	1	2	1
Тема 5. Віртуальні функції	5	1	2	2
Тема 6. Абстрактні класи	5	1	2	2
Тема 7. Стрічки	2	1	-	1
Тема 8. Робота з файлами	2	1	-	1
Тема 9. Організація графічного інтерфесу	2	1	2	1
Тема 10. Рисування на формі	5	1	2	2
Тема 11. Регулярні вирази	5	1	2	2
Змістовий модуль 2. Патерни проектування				
Тема 12. Патерни проектування	10	1	-	9
Тема 13. Патерн абстрактна фабрика (Abstract Factory)	9	1	2	4
Тема 14. Патерн будівельник (Builder)	9	1	2	4
Тема 15. Патерн прототип (Prototype)	5	1	-	4
Тема 16. Патерн одинак (Singleton)	5	1	-	4
Тема 17. Патерн адаптер (Adapter)	5	1	-	4
Тема 18. Патерн проксі (Proxy)	5	1	-	4
Тема 19. Патерн ітератор (Iterator)	5	1	-	4
Тема 20. Патерн спостерігач (Observer)	3	1	-	2
Змістовий модуль 3. Фреймворк Qt				
Тема 21. Фреймворк Qt	13	1	-	2
Тема 21. Слоти та сигнали	9	-	-	2
Тема 22. Бібліотека STL	4	-	-	4
Тема 23. Динамічні масиви	6	2	2	4
Всього	135	24	24	72

5. Теми лабораторних робіт

Таблиця 5.1

№	ТЕМА ЛАБОРАТОРНОЇ РОБОТИ	ЗМІСТ РОБОТИ	Обсяг, год.
1	2	3	4
1.	Інтерактивне середовище розробки Microsoft Visual Studio	Інструктаж з техніки безпеки. Ознайомитися з інтерактивним середовищем розробки Microsoft Visual Studio. Перша програма на C#. Навчитися створювати найпростіші Windows-додатки в середовищі Microsoft Visual Studio.	2
2.	Класи та об'єкти	Ознайомитися з особливостями ООП та його основними властивостями, навчитися створювати класи та об'єкти	2
3.	Наслідування	Ознайомитися з особливостями ООП та його основними властивостями, навчися використовувати ідеологію наслідування.	2
4.	Механізм віртуальних функцій	Ознайомитися з особливостями використання віртуальних функцій.	2
5.	Абстрактні класи	Навчитись використовувати абстрактні класи та інтерфейси	2
6.	Простір імен System.Drawing	Навчитись використовувати класи з простору імен System.Drawing для рисування на формі.	2
7.	Регулярні вирази	Навчитись використовувати класи з простору імен RegularExpression для операцій з текстом	2
8.	Патерни проектування	Навчитись використовувати породжуючі паттерни (абстрактна фабрика, будівельник та фабричний метод)	4
9.	Фреймворк Qt	Здійснити програмну реалізацію Web-агента для пошуку, фільтрації та збереження новин.	6
ВСЬОГО			24

Під час кожного лабораторного заняття студент виконує індивідуальне завдання з представленої в табл. 5.1 тематики.

Технічне забезпечення лабораторних робіт включає персональні комп'ютери, що мають вихід в мережу Інтернет.

Рекомендоване програмне забезпечення:

▲ Microsoft Windows 2000/2003/XP/Vista/7/8 або Linux

▲ Microsoft Office 2000/XP/2007/2010/Open Office

▲ Acrobat PDF Reader/Foxit PDF Reader

▲ Microsoft Visual Studio C#/C++ 2008/2010, Cygwin/Qt

6. Самостійна робота

6.1 Розподіл навчального часу на вивчення дисципліни

Число кредитів ECTS	Загальний обсяг дисципліни	Розподіл часу		Частка самостійної роботи, в %
		Аудиторні заняття	Самостійна робота	
4	120	48	72	60 %

6.1. Розподіл самостійної роботи студента

Враховуючи нормативні дані обрахунку годин самостійної роботи студента та обсяг навчальної роботи викладача обрахуємо сумарний об'єм самостійної роботи студента в семестрі:

$$S=24*0,5+24*0,5+4*6= 48 \text{ год.}$$

Перевищення передбаченої робочим навчальним планом самостійної роботи (72 год) становить $72-48=24$ год.

Самостійна робота є основним засобом засвоєння студентом навчального матеріалу у час, вільний від обов'язкових навчальних занять. Даний розподіл включає:

- вивчення лекційного матеріалу по темі (12 год - 0.5 год. на 1 год. занять);
- підготовка до лабораторних робіт (12 год - 0.5 год. на 1 год. занять);
- підготовку до заходів підсумкового контролю (24 год. - до 6 год. на 1 кредит ECTS).
- опрацювання питань навчальної програми, які не викладаються під час аудиторних занять (48 год - 3.0 год / 1 год лекції, яка не передбачається);

6.2. Завдання для самостійної роботи

Таблиця 6.1

№	НАЗВА ТЕМИ	Обсяг, год.
1.	Керований та некерований код	2
2.	Класи в мові C++	3
3.	Множинне наслідування в мові C++	3
4.	Віртуальні функції в мові C++	2
5.	Абстрактні класи в мові C++	2
6.	Поліморфізм в мові C++	3
7.	Перевантаження функцій в мові C++	3
8.	Віддалене відлагодження програми в середовищі Microsoft Visual Studio	2
9.	Патерн декоратор (Decorator)	2
10.	Патерн прототип (Prototype)	2
11.	Патерн одинак (Singleton)	2
12.	Патерн адаптер (Adapter).	2
13.	Патерн проксі (Proxy)	2
14.	Патерн ітератор (Iterator)	2
15.	Патерн фасад (Facade)	2
16.	Патерн медіатор (Mediator)	2
17.	Патерн відвідувач (Visitor)	2
18.	Методологія MVC	2
19.	Фреймворк Qt. Використання на інших платформах.	2
20.	Фреймворк Qt. Слоти та сигнали	2
21.	Бібліотека STL	2
22.	Динамічні масиви в C++ на прикладі класів	2
Разом		48

6.3. Оформлення звіту про самостійну роботу

Самостійна робота студента над засвоєнням навчального матеріалу з навчальної дисципліни може виконуватися у бібліотеці, навчальних кабінетах, комп'ютерних класах, а також у домашніх умовах.

Підсумком самостійної роботи є складання конспекту за темами, вказаними у п.6.2. Загальний обсяг конспекту визначається з умови повноти та якості викладеного матеріалу.

Конспект оформлюється на стандартному папері формату А4 або в зошиті. Конспект може бути рукописним або друкованим і виконується українською мовою.

Перевірка конспекту відбувається у терміни, спільно обумовлені студентом і викладачем.

7. Методи навчання

Лекції проводяться з використанням технічних засобів навчання і супроводжуються демонстрацією презентацій та прозорих плівок з основними теоретичними положеннями тем, алгоритмів, прикладами розв'язування задач.

Лабораторні заняття проводяться в комп'ютерному класі з використанням роздаткового матеріалу, методичних вказівок;

Модульні контрольні роботи проводяться в комп'ютерному класі з використанням інтерактивної тестуючої програми. Такий підхід забезпечує незалежність в оцінюванні рівня знань студента.

Після виконання лабораторної роботи формується звіт про її виконання та проводиться його захист.

8. Методи контролю

Для визначення рівня засвоєння студентами навчального матеріалу використовуються такі форми і методи контролю:

- 1) тестування після вивчення змістового модуля;
- 2) усне опитування під час захисту лабораторних робіт;
- 3) виконання додаткових індивідуальних завдань під час лабораторних робіт і консультацій;
- 4) відвідування лекцій;
- 5) конспект матеріалу в зошиті для тем, які виносяться на обов'язкове самостійне опрацювання студентами
- б) підсумковий залік.

9. Розподіл балів, які отримують студенти

Активність на лекціях – 10 балів (відвідування, відповіді на запитання).

Вчасне виконання, захист і здача звіту про виконання лабораторної роботи – 5 балів за 1 роботу (в сумі 45 балів).

Поточна перевірка знань у вигляді тестів — 25 балів.

Перевірка конспекту та результатів самостійної роботи — 10 балів.

Основними критеріями, що характеризують рівень компетентності студента при оцінюванні результатів поточного та підсумкового контролів з навчальної дисципліни, є:

- виконання всіх видів навчальної роботи, що передбачені робочою програмою навчальної дисципліни;
- глибина і характер знань навчального матеріалу за змістом навчальної дисципліни, що міститься в основних та додаткових рекомендованих літературних джерелах;
- вміння аналізувати явища, що вивчаються, у їх взаємозв'язку і розвитку;
- характер відповідей на поставлені питання (чіткість, лаконічність, логічність, послідовність тощо);

- вміння застосовувати теоретичні положення під час розв'язання практичних задач;
- вміння аналізувати достовірність одержаних результатів.

Таблиця 9.1

Поточне тестування та самостійна робота											
Змістовий модуль №1											
44											
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
Σ	2	7	3	7	3	7	3	7	2	2	4
T	1	1	1	1	1	1	1	1	1	1	1
Л	0	5	0	5	0	5	0	5	0	0	2
С	1	1	2	1	2	1	2	1	1	1	1

Таблиця 9.2

Поточне тестування та самостійна робота													Сума
Змістовий модуль №2										Змістовий модуль №3			
50										6			
	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	100
Σ	2	7	3	7	3	2	7	2	8	4	5	4	
T	1	1	1	1	1	1	1	1	1	1	1	1	
Л	0	5	0	5	0	0	5	0	5	2	2	2	
С	1	1	2	1	2	1	2	1	2	1	2	1	

T1, T2 ... T22 — теми змістових модулів, Т- теоретичні знання, Л- виконання завдань на лабораторних заняттях, С- самостійна робота.

Подальша деталізація балів наводиться у методичних вказівках для лабораторних занять та самостійної роботи.

Формою підсумкового контролю є залік, оцінювання рівня знань студентів здійснюється в межах від 0 до 100 балів (включно). Залік зараховується, якщо студент набрав не менше 60 балів та виконав усі лабораторні роботи.

Сума набраних балів та оцінка за 4-бальною шкалою оприлюднюються до початку екзаменаційної сесії у електронному журналі академічної групи. Конвертація 100-бальної шкали у 4-бальну здійснюється за табл. 9.2.

Таблиця 9.2

Шкали оцінювання (100-бальна та національна)

Сума балів	Оцінка за національною шкалою
90-100	Зараховано
82-89	
74-81	
64-73	
60-63	
35-59	Не зараховано з можливістю повторного складання
1-34	Не зараховано з обов'язковим повторним вивченням дисципліни

10. Методичне забезпечення

Методичне забезпечення навчального курсу «Об'єктно-орієнтоване програмування» включає:

- опорний конспект лекцій на паперовому та електронному носіях;
- завдання та методичні вказівки до виконання лабораторних робіт;
- завдання модульних контрольних робіт.

11. Рекомендована література

11.1. Базова

1. Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж., Приемы объектно-ориентированного проектирования. Паттерны проектирования; Издавництво «Питер», 2007 - 366с.
2. Медведев В.И.. Особенности объектно-ориентированного программирования на C++/CLI, C# и Java; Издавництво: РИЦ «Школа», 2010 – 444с.
3. Шилдт Г. C# 4.0. Полное руководство; пер. с англ. –М. : Издавництво “Вильямс”, 2011 – 1056 с.

11.2. Допоміжна

1. Нэш Т. C# 2010: ускоренный курс для профессионалов; пер. с англ. – М.: "Вильямс", 2010 – 592 с.
2. Прата С. Язык программирования C++. Лекции и упражнения; –М.: “Вильямс”, 2012. – 1248 с.
3. Страуструп Б. Язык программирования C++. – СПб.; М.: «Невский диалект» – «Изд-во Бином», 1999
4. Троелсен Э. Язык программирования C# 2010 и платформа.NET 4.0; пер. с англ. – М.: "Вильямс", 2011 – 1392 с.

12. Інформаційні ресурси

1. Будаї А. Дизайн-патерни - просто, як двері <http://designpatterns.andriybuday.com>
2. Биллиг В.А. Основы программирования на С# <http://www.intuit.ru/department/pl/csharp/24/>
3. Системи онлайн-освіти - <https://www.coursera.org>, <http://www.udacity.com>
4. Web-сайти періодичних видань - www.habrahabr.ru, www.prog.org.ru, <http://dou.ua>
5. Електронна бібліотека на сервері кафедри прикладної математики НУВГП за адресою: \\АМ\smb\5_course\

6. Бібліотеки:

НУВГП - м. Рівне, вул. Приходька, 75, 2-й корп. НУВГП тел.22-25-39;
Рівненська обласна універсальна наукова бібліотека:

33000, м. Рівне, пл. Короленка, 6,
Міжміський код 8(0362). Тел./факс: 22-10-63
E-mail: library@libr.rv.ua

Рівненська обласна бібліотека для юнацтва:

33027, м. Рівне, вул. Київська, 18
Міжміський код 8(0362). Тел. 23-02-98
E-mail: molody@ukr.west.net
URL: <http://libr.rv.ua>

Міська бібліотека міста Рівне:

33028, м. Рівне, вул. Корольова, 2
Міжміський код 8(0362) Тел. 5-42-86

Центральна бібліотека ЦБС міста Рівне:

33028, м. Рівне, вул. Київська, 44
Міжміський код 8(0362) Тел. 23-35-30
E-mail: library@icc.rv.ua
URL: <http://www.library.rv.ua>