

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**



**ФАКУЛЬТЕТ ЕКОНОМІКИ ТА ПІДПРИЄМНИЦТВА
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ЕКОНОМІЧНОЇ
КІБЕРНЕТИКИ**

Цирульник С. М.

ОБ'ЄКТНО-ОРІЄНТОВНЕ ПРОГРАМУВАННЯ

МЕТОДИЧНІ ВКАЗІВКИ

до підготовки та написання курсових проектів для студентів денної та заочної форми навчання першого бакалаврського освітнього рівня спеціальності 122 «Комп'ютерні науки та інформаційні технології»

Вінниця 2019

УДК 004.415(042.4) : 33 (072)

Методичні вказівки до підготовки та написання курсових проектів з дисципліни «Об'єктно-орієнтовне програмування» для студентів денної та заочної форми навчання першого бакалаврського освітнього рівня спеціальності 122 «Комп'ютерні науки та інформаційні технології» / Цирульник С. М.; Вінн. нац. аграр. ун-т. - Вінниця: ВНАУ, 2019. - 35 с.

Укладачі: Цирульник С. М., к.т.н., доцент кафедри комп'ютерних наук та економічної кібернетики

Рецензенти: Колесник І. С., к.т.н., доцент кафедри обчислювальної техніки Вінницького національного технічного університету
Шевчук О. Ф., к.ф.-м.н., доцент кафедри математики, фізики та комп'ютерних технологій ВНАУ

Затверджено до видання науково-методичною комісією ВНАУ (протокол № 3 від 18 жовтня 2019 р.) за поданням навчально-методичної комісії факультету економіки та підприємництва (протокол № 3 від 18 жовтня 2019 р.)

Методичні вказівки з курсу «Об'єктно-орієнтовне програмування» призначені до підготовки та написання курсових проектів для студентів денної та заочної форми навчання першого бакалаврського освітнього рівня спеціальності 122 «Комп'ютерні науки та інформаційні технології».

Методичні вказівки містять загальні вимоги до виконання та оформлення курсових проектів, спрямованих на вирішення науково-практичних завдань з основ сучасної технології створення складних програмних продуктів на базі принципів об'єктно-орієнтованого методу з урахуванням сучасних вимог у відношенні до надійності, якості інтерфейсу та їх ефективності.

ВСТУП

Сучасні технології програмування переживають період бурхливого розвитку. Причинами цього явища є зростання потужності комп'ютерної техніки, її здешевлення, створення всесвітньої мережі Internet, виникнення все нових і нових сфер застосування комп'ютерів та потреба у різноманітному програмному забезпеченні.

Сучасне програмне забезпечення стає щораз складнішим. Зручний у використанні інтерфейс, розвинені можливості, керованість програми подіями, нові нетрадиційні сфери застосування – все це зумовлює ускладнення програмного забезпечення. Сьогодні, щоб побудувати сучасну програму, не достатньо просто об'єднати в послідовність певні машинні інструкції, оператори мови високого рівня чи навіть набори процедур та модулів. Головним стало питання розробки виразної структури програми, придатної до легкої модифікації, вільної від помилок, стійкої до змін.

Об'єктно-орієнтована технологія створення програмного забезпечення була задумана і розроблена як інструмент подолання складності. Вона успадкувала всі найкращі надбання структурного та модульного програмування, використавши їх для реалізації ряду принципово нових підходів до проектування програмного забезпечення. Головним завданням об'єктно-орієнтованого підходу є забезпечення способу структурування програми та керування складними взаємозв'язками між великою кількістю компонентів системи..

Методичні вказівки містять загальні вимоги до написання та оформлення курсових проектів з дисципліни «Об'єктно-орієнтовне програмування» для студентів денної та заочної форми навчання першого бакалаврського освітнього рівня спеціальності 122 «Комп'ютерні науки та інформаційні технології». У методичних вказівках визначається мета, завдання та послідовність виконання курсових проектів, наводяться рекомендації щодо вибору теми і роботи з літературними джерелами, вимоги до структури, змісту і оформлення роботи.

ЗМІСТ

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ	4
2. ВИБІР ТЕМИ КУРСОВОГО ПРОЕКТУ	6
3. ВИВЧЕННЯ ЛІТЕРАТУРИ ТА ІНШИХ МАТЕРІАЛІВ.....	6
4. СТРУКТУРА І ЗМІСТ КУРСОВОГО ПРОЕКТУ. СКЛАДАННЯ ПЛАНУ ..	6
4.1. Аналіз технічного завдання	9
4.2. Обґрунтування алгоритму й структури програми	9
4.3. Розробка програми	10
4.3.1. Загальні правила	10
4.3.2. Розробка системи класів	11
4.3.3. Розробка методів	13
4.3.4. Створення об'єктів і розробка головної програми	14
4.3.5. Опис файлів даних та інтерфейсу програми	14
4.4. Тестування програми та результати її виконання	14
5. ЗАГАЛЬНІ ВИМОГИ ДО КУРСОВОГО ПРОЕКТУ ТА ЇЇ ОФОРМЛЕННЯ	15
6. ПОДАННЯ НА ПЕРЕВІРКУ, ПІДГОТОВКА ДО ЗАХИСТУ І ЗАХИСТ КУРСОВОГО ПРОЕКТУ	21
6.1. Підготовка до захисту курсового проекту	21
6.2. Захист курсового проекту та критерії оцінки	21
7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА	24
Додаток А. Орієнтовна тематика курсових проектів з дисципліни «Об'єктно-орієнтовне програмування»	25
Додаток Б. Зразок оформлення титульної сторінки	28
Додаток В. Зразок бланка рецензії на курсовий проект	29
Додаток Г. Зразок завдання курсового проекту.....	30
Додаток Д. Приклади оформлення бібліографічного опису у списку джерел до курсового проекту.....	31

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Дане видання призначене для допомоги студентам при виконанні курсового проекту з дисципліни «Об'єктно - орієнтоване програмування».

Мета виконання курсового проекту – набуття практичних навичок в галузі практичного застосування об'єктно - орієнтованого підходу до проектування та реалізації програмних систем.

У процесі виконання курсового проекту студент повинен вміти: самостійно вибрати середовище проектування програмних засобів, проектувати, розробляти, налагоджувати та супроводжувати сучасне програмне забезпечення.

Під час виконання курсового проекту студент повинен продемонструвати:

- вміння збирати та аналізувати відповідні матеріали про об'єкт дослідження, використовуючи сучасні джерела інформації, включаючи Інтернет ресурси;
- спроможність проводити необхідні обґрунтування для розробки програмних засобів різного призначення, тощо;
- здатність доводити розв'язання поставленої задачі до логічного кінця;
- вміння аналізувати отримані результати та робити з них висновки.

Виконання курсового проекту складається з таких етапів:

1. Вибір теми, визначення мети, конкретних завдань, предмета та об'єкта дослідження.
2. Обговорення теми та затвердження її керівником від кафедри.
3. Складання плану курсового проекту, пошук та вивчення необхідних літературних джерел.
4. Вибір та обґрунтування проектних рішень, розробка та кодування програмної системи, тестування системи.
5. Написання тексту курсового проекту та її оформлення.
6. Подання курсового проекту на перевірку.
7. Доопрацювання (у разі необхідності).
8. Підготовка до захисту та захист курсового проекту.

2. ВИБІР ТЕМИ КУРСОВОГО ПРОЕКТУ

Вибір теми курсового проекту є першим етапом. Студенти вибирають тему курсового проекту самостійно відповідно до тематики, визначеної кафедрою. Крім того, передбачена можливість вибору студентом вільної теми, узгодженої зі змістом дисципліни та затвердженої на засіданні кафедри.

Орієнтовну тематику курсових проектів з дисципліни «Об'єктно - орієнтоване програмування» наведено в Додатку А.

При виборі теми враховуються особисті здібності, наукові інтереси, характер попередніх та майбутніх досліджень кожного студента. Крім того, необхідно звернути увагу на актуальність і значущість теми, наявність та доступність необхідного теоретичного та практичного матеріалу, можливість проведення відповідних розрахунків на підставі матеріалів дослідження.

3. ВИВЧЕННЯ ЛІТЕРАТУРИ ТА ІНШИХ МАТЕРІАЛІВ

Успішному написанню курсового проекту допомагає робота з літературними джерелами. Ознайомлення найкраще починати з підручників та навчальних посібників, де матеріал викладено послідовно та систематизовано, але при цьому не можна обмежуватися підручниками. Необхідно, щоб студент використовував і періодичні професійні видання - збірники наукових праць, журнали, оглядову інформацію. Велику цінність для написання курсових проектів мають наукові монографії.

У процесі підбору літератури студент здійснює пошук інформації за всіма доступними джерелами з вибраної теми. Вивчає стан вивчення даної теми в науковій літературі та визначає конкретні напрями дослідження. Доцільно при вивченні літератури робити виписки, систематизуючи відібрані літературні джерела за основними напрямками роботи і, водночас, складати список використаних літературних джерел.

Обов'язковим є вивчення науково-практичного матеріалу спеціалізованих періодичних видань, які відрізняються своєю змістовністю, конкретністю, сучасністю та актуальністю; побудовані згідно з новими вимогами та враховують зміни в законодавстві; містять передовий практичний досвід.

4. СТРУКТУРА І ЗМІСТ КУРСОВОГО ПРОЕКТУ. СКЛАДАННЯ ПЛАНУ

Після вибору теми курсового проекту та погодження її із керівником студент складає план у письмовому вигляді, попередньо ознайомившись із літературою,

що дає уяву про зміст і структуру вибраної теми, її важливі моменти, проблеми, що підлягають вирішенню.

Визначивши тему та вивчивши відповідну літературу, студент разом з керівником має скласти план курсового проекту.

План — це основа роботи, тому його складання є одним із найважливіших етапів її підготовки. План повинен відобразити напрями дослідження обраної теми, логічний зв'язок між її окремими складовими частинами, проблемну постановку окремих питань. Із складеного плану видно, наскільки студент засвоїв матеріал обраної теми, як він зрозумів проблему в цілому і зумів виділити суттєві, головні напрями дослідження. Від правильно складеного плану багато в чому залежить і кінцевий результат виконаної роботи.

План повинен складатися з таких розділів:

- вступ;
- формулювання вимог до програми;
- проектування програми;
- кодування програми;
- тестування програми
- висновки (кінцева частина);
- література;
- додатки.

Формулювання вимог до програми є самостійним етапом роботи. На етапі формулювання вимог студент повинен:

- 1) сформулювати призначення програми;
- 2) визначити перелік функцій, які виконує програма;
- 3) запропонувати інтерфейс користувача, який надає можливість останньому дістатися цих функцій.

На *етапі проектування* визначається архітектурний стиль, об'єктна структура програми, а також формат похідних, вихідних та внутрішніх даних. Співвідношення між класами та об'єктами відображується за допомогою діаграм, бажано на мові UML. Під час проектування можна розробити план функціонального тестування, та необхідні тести. Завершується проектування створенням проектної специфікації програми.

На етапі *кодування* створюється похідний код програми. Він повинен бути гідним чином прокоментований і повністю задовольняти вимогам до похідного коду. Увесь похідний код програми має бути наведений у додатку до пояснювальної записки. У тексті основних розділах записки можливо наводити лише невеликі за розміром окремі фрагменти коду.

Розрізняють два основних різновиду *тестування*: функціональне та модульне. Функціональне тестування перевіряє, як програма виконує свої

функції в умовах коректних та некоректних дій користувача. Воно планується на етапі проектування і виконується у міру того, як кодуються окремі функції програми і перед захистом програми. Модульне тестування робиться одночасно з кодуванням і має за мету оперативну перевірку окремих частин програми. Виконання модульного тестування бажано, але не є обов'язковим.

Загальний обсяг курсового проекту повинен складати 25-35 сторінок машинописного тексту на аркушах формату А4 (210 x 297). Список літератури і додатки до вказаного обсягу не входять.

Структура курсового проекту:

1. Титульний аркуш (див. Додаток Б).
2. Рецензія (див. Додаток В).
3. Анотація.
4. Зміст.
5. Вступ.
6. Основна частина (3 розділи).
7. Висновки.
8. Література (див. Додаток Д).
9. Додатки (за наявністю).

Анотація призначена для ознайомлення із змістом курсового проекту. Вона має бути стислою, інформативною і містити відомості, які характеризують виконану роботу. Анотацію слід розміщувати безпосередньо за індивідуальним завданням, починаючи з нової сторінки.

В анотації дається коротка характеристика основного змісту курсового проекту та одержаних результатів дослідження. При цьому використовуються переважно прості синтаксичні конструкції, характерні стилю ділових документів та стандартизована термінологія.

Анотація завершується наведенням ключових слів - сталих термінологічних словосполучень, які визначають смислове навантаження курсового проекту. Ключові слова подаються у називному відмінку, друкуються в рядок, через кому, їх загальна кількість повинна бути не більше десяти слів.

Зміст розміщують з нової сторінки. Слово «ЗМІСТ» записують у вигляді заголовка симетрично до тексту. Між назвою і текстом змісту, як і інших структурних частин проекту, пропускають 1 рядок. Зміст курсового проекту містить найменування та номери початкових сторінок усіх розділів та підрозділів роботи. Зміст має включати усі заголовки, які є в роботі, починаючи зі вступу і закінчуючи додатками.

Викладення матеріалу курсового проекту починається зі вступу (2 с.).

У вступі коротко викладають: оцінку сучасного стану проблеми, відмічаючи практично розв'язані задачі, прогалини знань, що існують у даній галузі,

провідні фірми та провідних вчених і фахівців даної галузі; світові тенденції розв'язання поставлених задач; актуальність даної роботи та підставу для її виконання; мету роботи та галузь застосування; взаємозв'язок з іншими роботами.

4.1. Аналіз технічного завдання

Основною метою даного розділу пояснювальної записки є аналіз вимог завдання на курсовий проект й формулювання додаткових технічних вимог, які безпосередньо впливають з нього, та мети дослідження.

На основі аналізу потрібно конкретизувати основні параметри програми, її можливості та функції, засоби їх реалізації, інтерфейс із користувачем, середовище, в якому програма може працювати.

Обсяг розділу — 1..2 сторінки.

4.2. Обґрунтування алгоритму й структури програми

Виходячи із поставленого завдання та технічних вимог до програми, потрібно розробити алгоритм її роботи. Він може бути відображений у вигляді UML-діаграми (при необхідності блок-схеми, оформленій згідно з вимогами стандартів, або словесним описом, що включає послідовність узагальнених операцій та переходів між ними).

Такий опис може мати ієрархічну структуру, де спочатку записують більш загальні операції, які потім розшифровують на нижчому ієрархічному рівні.

Опис алгоритму при використанні об'єктно-орієнтованого підходу проектування програмного забезпечення зручніше проводити в термінах об'єктів, що взаємодіють між собою за допомогою UML діаграм відношень, повідомлень і подій. У деяких випадках алгоритм може розпадатися на кілька незалежних одна від одної послідовностей операцій, що реагують на події та керують опрацюванням повідомлень.

Для систем, що працюють у реальному часі, важливим є часова послідовність опрацювання повідомлень і подій. Для них, крім опису алгоритму, застосовують ще й часові UML-діаграми, що характеризують послідовність виконання різних процесів, та їх взаємодію з апаратними засобами.

У загальному, спосіб опису алгоритму вибирають, виходячи з конкретної задачі, однак він повинен бути повним, несуперечливим і достатнім для розуміння суті та ілюстрації запропонованого способу її вирішення та принципу роботи спроектованої програми.

Аналізуючи отриманий алгоритм, потрібно визначити в ньому основні

структурні частини, які дозволяють здійснити декомпозицію (розбиття) проекту на завершені програмні одиниці (модулі, програми, бібліотеки). Це в більшості випадків спрощує як процес розробки, так і відлагодження програмного проекту.

Найчастіше окремі методи (функції), класи та змінні групують за функціональним призначенням та розміщують в окремих програмних одиницях (модулях), які можна окремо компілювати. Для програмування на мові C++ кожен із модулів, як правило, розбивають на два файли: заголовний файл із розширенням *.h, що містить інтерфейсну частину модуля, та файл реалізації модуля з розширенням *.cpp.

Крім модулів, потрібна головна програма, яка їх викликає. Для випадку, якщо завданням передбачено розробку повноцінної програми, то ця головна програма й реалізує необхідний алгоритм роботи. Якщо ви розробляєте лише набір модулів або бібліотеку функцій (наприклад, реалізуєте побудову графіків на основі даних, сформованих у програмі користувача, або створюєте бібліотеку для реалізації різноманітних операцій з новим типом даних, таким як комплексні числа), то така головна програма буде лише тестовою, призначеною для відлагодження, перевірки та демонстрування можливостей модулів.

Згідно з указаною вище схемою потрібно провести декомпозицію проекту та відобразити її в даному розділі.

Обсяг розділу — 2..4 сторінки.

4.3. Розробка програми

4.3.1. Загальні правила

Цей розділ є основним у курсовому проекті. Його обсяг – не менше 8.. сторінок.

У залежності від обраного стилю програмування, можливі різні підходи до розробки програми. У найпростіших випадках вона йде послідовно, алгоритм програми реалізується за допомогою послідовності операторів, включно з умовними та операторами циклу.

У структурному програмуванні застосовується ієрархічний підхід, у якому окремі структурні одиниці перетворюються на функції, що можуть приймати параметри та повертати результат. Опис функції в такому випадку виконується один раз у певному місці програми, а використовується вона багато разів, при цьому лише змінюються параметри, і не потрібно щоразу повторювати одні й ті ж оператори, які включені у функцію.

Найвищу продуктивність роботи при розробці складних програмних

продуктів дає об'єктно-орієнтований підхід, який завдяки принципам інкапсуляції, успадкування та поліморфізму дає можливість виділити всі спільні та відмінні риси окремих елементів поставленого завдання і записати їх у програмі в найлаконічнішій формі.

Розробка програми при об'єктно-орієнтованому підході розбивається на розробку класів та глобальних типів даних, реалізацію методів, дружніх та глобальних функцій, створення об'єктів та організацію взаємодії між ними в головній програмі для досягнення поставленої мети.

4.3.2 Розробка системи класів

Після того, як виконана декомпозиція програми (розроблена її структура) і відомі алгоритми розв'язку задач, можна приступати до реалізації наміченого плану. Потрібно чітко уявити, з якими даними буде працювати програма (чи окремий модуль) та які дії над ними доведеться виконувати. Дані та дії над ними утворюють об'єкт. Також можна уявити програму як набір об'єктів (реально існуючих фізичних об'єктів або абстрактних понять, що існують лише в нашій уяві), які взаємодіють між собою. Наступним етапом буде поділ об'єктів на групи, що мають хоча б щось (дані, властивості, функціональність). У більшості випадків об'єкти будуть більше чи менше подібні. Деколи в групах можна буде виділити підгрупи і т. д., аж до складного ієрархічного дерева. У кожній групі виділяється найзагальніші властивості, притаманні всім без винятку об'єктам у групі та дії над ними. Отже, отримаємо базовий клас для цієї групи об'єктів, даними якого будуть загальні параметри об'єктів групи, а методами — дії, які можна проводити над даними будь-якого об'єкта групи. Далі для кожної з підгруп (у межах виділеного класу об'єктів) додаємо притаманні лише їй властивості, як поля нового, породженого від базового класу. Додаткові дії, які властиві кожній підгрупі, стають методами породжених класів, а параметри, яких не мали об'єкти базового класу, — даними породжених (похідних) класів. Породження класів продовжується до тих пір, поки не будуть описані всі параметри об'єктів та дії, які необхідно над ними здійснювати для розв'язку поставлених задач. Описана послідовність повторюється для кожної з Групи об'єктів у програмі.

Для прикладу розглянемо програму, яка повинна грати в шахи. У ній можна виділити такі дві Групи об'єктів: перша — реалізує графічне відображення на екрані шахової дошки та фігур на ній, друга — реалізує логіку роботи програми — генерує ходи, перевіряє їх правильність, слідкує за часом і т. д. Розглянемо першу групу. З чого вона складається? Сюди входять різноманітні видимі елементи: дошка, клітинки на ній, різні фігури, позначення рядів на дошці. Які в

них параметри? Це — координати, розміри, колір для фігур, для клітинки — ще тип фігури, яка на ній знаходиться, для дошки — ще масив клітинок на ній і т. д. Які дії повинні виконувати об'єкти? Вони повинні реагувати або не реагувати на натискання мишкою чи на клавіатуру, відображатися, зникати, деякі ще повинні переміщуватись, причому по-різному. Що ж є спільним для них усіх? Серед параметрів — напевне, координати центра, ще колір і розмір по горизонталі та вертикалі, з дій — кожен із об'єктів повинен уміти створюватися, знищуватися та відображати себе на екрані. Таким чином, базовий клас буде містити такі поля даних: координати «x», «y», «ширина», «висота», «колір»; методи — конструктор, що створює об'єкт, деструктор, що його знищує, та метод «відобрази себе». Оце й усе! Але описане лише один раз (!) для всіх видимих об'єктів. Далі можна розділити видимі об'єкти на пасивні та активні. Пасивними будуть написи на дошці, що позначають назви рядів, активними — всі решта, вони повинні мати метод, який змушує їх реагувати на натискання мишкою. Тепер виділимо ще підклас фігур, що породжується від класу «активні видимі елементи», який має додатковий метод «перемісти мене» та поле даних або метод, що визначає допустимі ходи для даної фігури. Так, дошка містить 64 клітинки, а кожна клітинка — може містити якусь фігуру. Тому до даних цього класу, також породженого від «активних видимих елементів», слід додати ще поле даних (напевне, вказівник), який містить інший об'єкт(-и) класу — «видимі елементи». Також необхідні методи, які ці об'єкти «вставляють» у клас «Група» або видаляють з «Групи». Метод «відобрази себе» для «групи» завдяки поліморфізму можна доповнити відображенням об'єктів «групи» після того, як відобразилася сама «група». Далі від «групи» можна успадкувати нові класи: «шахова дошка», «шахове поле», а від «шахового поля» — «поле останнього рядка», яке крім методів свого попередника може ще й перетворювати «пішака» у «королеву». Цей, далеко не повний, приклад дає можливість зрозуміти лише основний підхід до проектування класів. Пофантазувавши трохи, можна отримати досить елегантну й, головне, завершену ієрархію об'єктів, які будуть уміти робити все, що від них може вимагати головна програма для розв'язування поставлених задач. Основне — кожна спільна для кількох об'єктів дія (і спільний елемент даних) описана лише один раз у базовому класі і більше ніде її не потрібно описувати повторно. Якщо деякі дії передбачають виконання операцій над кількома об'єктами і при цьому необхідно мати безпосередній доступ до закритих даних класу, то такі дії робимо дружніми функціями для того класу, над даними якого вони працюють. Деякі типи даних, які не доцільно робити класом, оскільки вони не мають власної функціональності, але які неодноразово використовуються в програмі (наприклад, масиви фіксованої довжини, списки чи структури) можемо описати, використавши `typedef` помістити в заголовок

одного з модулів.

Підрозділ «Розробка системи класів» відображає основну роботу, яка виконується при об'єктно-орієнтованому програмуванні, в результаті чого отримуються описи всіх класів з даними та методами, причому для методів уже задані їх параметри та тип результату, який вони повертають. Тобто, з точки зору інтерфейсу, робота над класами на цьому етапі вже завершена. Залишилося тільки «навчити» методи виконувати свою роботу. Глобальні функції, які не є методами ні одного з класів, також поміщаються в один із модулів, а їх інтерфейсна частина описується в цьому підрозділі.

4.3.3 Розробка методів

Після того, як з інтерфейсною частиною «закінчено», залишилося лише розробити реалізацію методів, дружніх та глобальних функцій. Це завдання є досить простим, якщо на попередньому етапі було правильно спроектовано систему класів та детально пророблено інтерфейсні частини. Реалізація будь-якої функції (чи методу) розміщується в *.cpp - частині модуля, на відміну від інтерфейсу, яка записується у *.h- файл. Потрібно лише акуратно записати ті дії, які має виконувати метод чи функція, не забуваючи користуватися вже готовими функціями та методами замість того, щоб увесь час змінювати дані за допомогою одних і тих же послідовностей операцій. Так, метод «перемісти мене» (див. приклад у попередньому пункті) повинен лише змінити координати активного видимого елемента, а не намагатися перемалювати його самостійно — метод «відобрази себе» зробить це набагато краще. Аналогічно метод «відобрази себе» для «групи» (шахової дошки чи клітинки), який повинен відображати об'єкти Групи після того, як відобразилася сама група, реалізується з використанням поліморфізму та успадкування. При цьому додається лише один-два оператори, оскільки і група, і видимі елементи вже самі вміють відображатися, необхідно лише «сказати» їм, щоб вони це зробили один за одним. У методі «відобрази себе» групи спочатку викликається метод «відобрази себе» базового класу («видимий елемент»), який відображає саму дошку або клітинку, а далі — методи «відобрази себе» для всіх видимих елементів, що входять до групи. Для клітинки Це буде виклик «відобрази себе» для фігури, що стоїть на клітинці, а для дошки — виклик «відобрази себе» для всіх позначок та клітинок, які, у свою чергу, відобразять фігури, що на них знаходяться. У даному підрозділі слід описати принцип дії основних методів та відмінності поліморфних (віртуальних) методів від методів базових класів.

4.3.4 Створення об'єктів і розробка головної програми

У цьому підрозділі слід описати процес створення самих об'єктів у програмі, функціональність та дані яких розроблені в попередніх підрозділах. Якщо кількість об'єктів, що будуть створені наперед невідома, або обсяг пам'яті, яку вони займають, є досить значним, тоді пам'ять для об'єктів слід виділяти «динамічно» з використанням оператора `new`. Стосовно розглянутого прикладу із шахами, даний підрозділ, очевидно буде описувати, як створюється шахова дошка з 32 фігурами на ній та як перетворюються вибрані партнерами ходи в повідомлення шаховим фігурам. Створивши об'єкти, можемо реалізувати певну послідовність дій або схему взаємодії між об'єктами, яка призводить до вирішення поставленого завдання. Опис цієї послідовності дій чи схеми взаємодії, можливо, з прикладами для реальних вхідних даних чи повідомлень і є основою даного підрозділу. Такий опис повинен бути достатнім для розуміння всіх деталей реалізації алгоритму та можливих випадків, що зустрічаються при роботі програми.

4.3.5 Опис файлів даних та інтерфейсу програми

Якщо програма використовує файли як джерело вхідних даних або для зберігання проміжних чи кінцевих результатів роботи, то в даному підрозділі слід навести опис формату цих файлів. Він може бути виконаний у текстовому, табличному чи графічному вигляді. Можна додавати до нього приклади реальних файлів із даними програми. Особливо важливий даний підрозділ для програм, що працюють із базами даних, адже структура таблиць бази даних, перелік, типи полів, засоби взаємодії з базою даних, перетворення даних є основою таких програм. Для інтерактивної програми з розвинутою системою меню та діалогових вікон у цьому підрозділі слід описати призначення елементів меню, роботу з ними, параметри, що вибираються в діалогових вікнах тощо. У цьому ж розділі слід описати інтерфейс програм, які працюють з параметрами командного рядка. Якщо темою роботи є розробка модулів або бібліотеки, то слід детально описати порядок їх використання, навести в алфавітному порядку всі доступні для користувача глобальні типи даних і змінні, функції та класи.

4.4. Тестування програми та результати її виконання

У даному розділі треба описати методику тестування програми, тестові дані та навести результати роботи програми. Якщо програма працює в графічному режимі, то слід роздрукувати копію графічного вікна програми. Якщо

результатом роботи програми є текстовий файл, то необхідно вивести вміст цього файлу. Для програм з розвинутою системою діалогових вікон і меню слід обмежитися друком лише найсуттєвіших результатів, які демонструють правильну роботу програми, а не передруковувати весь екран для кожного відкритого пункту меню. Перелік усіх пунктів меню в такому випадку та вміст неосновних діалогових вікон можна подати в текстовому вигляді. Якщо для відображення роботи програми необхідна значна кількість роздруківок, то їх можна подати в додатках. Обсяг розділу — 1..5 сторінок. У розділі потрібно зробити висновок, який підтверджує (або заперечує) працездатність програми.

Кожен із розділів курсового проекту слід завершувати коротким узагальнюючим висновком, який не виділяється окремим структурним елементом, на відміну від загальних висновків.

У висновках до курсового проекту в реферативній формі повинні бути описані результати, отримані студентом на кожному з етапів виконання проекту, а також висновків щодо досягнення мети курсового проекту, перспективи розвитку даної галузі тощо. Висновки пишуться на 1-2 сторінки і як розділ не нумеруються.

За результати дослідження, розрахунків та зроблені на цій основі висновки відповідальність несе студент - автор курсового проекту.

Список використаної літератури містить складений за чинними правилами перелік використаних літературних та інформаційних джерел (Додаток Д). Перелік посилань на літературні джерела подається у порядку посилання на них у тексті або в алфавітному порядку.

Додатки можуть містити матеріал, який не може бути послідовно розміщений в основній частині записки через великий обсяг але є необхідним для фахівців даної галузі.

У додатки можуть бути включені: додаткові ілюстрації або таблиці; оригінали фотографій, проміжні математичні докази, формули, розрахунки; інструкції, методики, опис комп'ютерних програм, розроблених у процесі виконання роботи та ін.

5. ЗАГАЛЬНІ ВИМОГИ ДО КУРСОВОГО ПРОЕКТУ ТА ЇЇ ОФОРМЛЕННЯ

Пояснювальну записку виконують на аркушах білого паперу формату А4 (210×297 мм). Пояснювальна записка має друкуватися з одного боку листа через 1,5 інтервали чорнилами чорного кольору.

Пояснювальна записка оформлюється відповідно до вимог ГОСТ 7.32-91 «Отчет о научно-исследовательской работе. Структура и правила оформления»

та СТП 15-96.

Документи на програму оформлюються у відповідності до вимог ЕСПД. Загальні вимоги до оформлення програмних документів викладені у ГОСТ 19.104-78, 19.105-78, 19.106-78.

Вимоги до розробки окремих програмних та експлуатаційних документів представлені у таких стандартах: до тексту програми - у ГОСТ 19.401-78; до опису програми - у ГОСТ 19.402-78; до інструкції користувача - у ГОСТ 19.505.

Список використаних джерел та посилання на нього у тексті виконуються у відповідності з ДСТУ 8302:2015 та міжнародного стилю АРА. Заголовки структурних елементів «РЕФЕРАТ», «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ», «ВСТУП», «ВИСНОВКИ», «РЕКОМЕНДАЦІЇ», «ПЕРЕЛІК ПОСИЛАНЬ», «ЛІТЕРАТУРА» не нумерують, а їх назви правлять за заголовки структурних елементів.

Заголовки структурних елементів пояснювальної записки і заголовки розділів слід розташовувати посередині рядка і друкувати великими літерами без крапки в кінці, не підкреслюючи.

Розділи і підрозділи повинні мати заголовки. Пункти і підпункти можуть мати заголовки. Заголовки підрозділів, пунктів і підпунктів пояснювальної записки слід починати з абзацного відступу і друкувати маленькими літерами, крім першої великої, не підкреслюючи, без крапки в кінці. Якщо заголовок складається з двох і більше речень, їх розділяють крапкою. Перенесення слів у заголовку розділів не допускається.

Відстань між заголовком і подальшим чи попереднім текстом має бути не менше ніж два рядки. Відстань між основами рядків заголовку, а також між двома за-головками приймають такою, як у тексті.

Абзацний відступ повинен бути однаковим впродовж усього тексту пояснювальної записки і дорівнювати 1,27 см.

Не допускається розміщувати назву розділу, підрозділу, а також пункту й підпункту в нижній частині сторінки, якщо після неї розміщено тільки один рядок тексту.

Сторінки пояснювальної записки, слід нумерувати арабськими цифрами, додержуючись наскрізної нумерації впродовж усього тексту пояснювальної записки. Номер сторінки проставляють у правому верхньому куті сторінки без крапки в кінці. Титульний аркуш включають до загальної нумерації сторінок пояснювальної записки. Номер сторінки на титульному аркуші не проставляють. Ілюстрації й таблиці, розмішені на окремих сторінках, включають до загальної нумерації сторінок пояснювальної записки. Розділи, підрозділи, пункти, підпункти пояснювальної записки слід нумерувати арабськими цифрами. Розділи повинні мати порядкову нумерацію в межах викладення суті

пояснювальної записки і позначатися арабськими цифрами без крапки, наприклад, 1, 2, 3 і т. д. Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, відокремлених крапкою. Після номера підрозділу крапку не ставлять, наприклад 1.1, 1.2 і т.д.

Пункти повинні мати порядкову нумерацію в межах кожного розділу або підрозділу. Номер пункту складається з номера розділу і порядкового номера пункту або з номера розділу, порядкового номера підрозділу та порядкового номера пункту, відокремлених крапкою. Після номера пункту крапку не ставлять, наприклад, 1.1, 1.2, або 1.1.1, 1.1.2 і т. д. Якщо текст поділяють тільки на пункти, їх слід нумерувати, за винятком додатків, порядковими номерами.

Номер підпункту складається з номера розділу, порядкового номера підрозділу, порядкового номера пункту і порядкового номера підпункту, відокремлених крапкою, наприклад, 1.1.1.1, 1.1.1.2, 1.1.1.3 і т. д. Якщо розділ, не маючи підрозділів, поділяється на пункти і далі - на підпункти, номер підпункту складається з номера розділу, порядкового номера пункту і порядкового номера підпункту, відокремлених крапкою, наприклад, 1.1.3, 1.2.1 і т. д. Після номера підпункту крапку не ставлять. Якщо розділ або підрозділ складається з одного пункту або пункт складається з одного підпункту, його нумерують.

Ілюстрації (креслення, рисунки, графіки, схеми, діаграми, фотознімки) слід розміщувати безпосередньо після тексту, де вони згадуються вперше, або на наступній сторінці. На всі ілюстрації мають бути посилання у тексті. Якщо ілюстрації створені не автором пояснювальної записки, необхідно при поданні їх дотримуватись вимог чинного законодавства про авторські права. Креслення, рисунки, графіки, схеми, діаграми, розміщені у пояснювальній записці, мають відповідати вимогам стандартів «Єдиної системи конструкторської документації» та «Єдиної системи програмної документації». Ілюстрації можуть мати назву, яку розміщують під ілюстрацією. За необхідності під ілюстрацією розміщують пояснювальні дані (підрисунковий текст). Ілюстрація позначається словом «Рисунок ___», яке разом з назвою ілюстрації розміщують після пояснювальних даних, наприклад, «Рисунок 3.1 – Схема розміщення». Ілюстрації слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком ілюстрацій, наведених у додатках. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, відокремлених крапкою, наприклад, рисунок 3.2 – другий рисунок третього розділу. Рисунки вставляють у текст із графічного файлу. Забороняється використання рисунків WordArt та вбудованого редактору векторних рисунків.

Таблиці. Цифровий матеріал, як правило, оформлюють у вигляді таблиць.

Горизонтальні та вертикальні лінії, які розмежують рядки таблиці, а також лінії зліва, справа і знизу, що обмежують таблицю, можна не проводити, якщо їх відсутність не заважає користування таблицею. Таблицю слід розташовувати безпосередньо після тексту, у якому вона згадується вперше таким чином, щоб її можна було читати без повороту переплетеного блоку чи з поворотом за годинниковою стрілкою, або на наступній сторінці. Перед заголовком таблиці та після нього, а також після таблиці пропускають по одному рядку. На всі таблиці мають бути посилання в тексті. При перенесенні таблиці на інший аркуш (сторінку) назву вміщують тільки над її першою частиною. Таблицю з великою кількістю граф можна ділити на частини і розміщувати одну частину під іншою в межах одної сторінки. Якщо рядки або графи таблиці виходять за формат сторінки, то в першому випадку в кожній частині таблиці повторюють її головку, в другому випадку - боковик.

Таблиці слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком таблиць, що наводяться у додатках.

Номер таблиці складається з номера розділу і порядкового номера таблиці, відокремлених крапкою, наприклад, таблиця 2.1 – перша таблиця другого розділу. Таблиця може мати назву, яку друкують малими літерами (крім першої великої) і вміщують над таблицею. Назва має бути стислою і відбивати зміст таблиці.

Якщо рядки або графи таблиці виходять за межі формату сторінки, таблицю поділяють на частини, розміщуючи одну частину під одною, або поруч, або переносючи частину таблиці на наступну сторінку, повторюючи в кожній частині таблиці її головку і боковик.

При поділі таблиці на частини допускається її боковик замінити відповідно номерами граф чи рядків, нумеруючи їх арабськими цифрами у першій частині таблиці.

Слово «Таблиця ____» вказують один раз зліва над першою частиною таблиці, над іншими частинами пишуть: «Продовження таблиці ____» з зазначенням номера таблиці.

Заголовки граф таблиці починають з великої літери, а підзаголовки – з малої, якщо вони складають одне речення з заголовком. Підзаголовки, що мають самостійне значення, пишуть з великої літери. В кінці заголовків і підзаголовків таблиць крапки не ставлять. Заголовки і підзаголовки граф указують в однині.

Ставити лапки замість цифр, марок, знаків, математичних і хімічних символів, які повторюються, не слід. Якщо цифрові або інші дані в якому-небудь рядку таблиці не подають, то в ньому ставлять прочерк.

Переліки, за потреби, можуть бути наведені всередині пунктів або підпунктів. Перед переліком ставлять двокрапку. Перед кожною позицією переліку слід

ставити малу літеру української абетки з дужкою, або, не нумеруючи - дефіс (перший рівень деталізації). Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації). Переліки першого рівня деталізації друкують малими літерами з абзацного відступу, другого рівня - з відступом відносно місця розташування переліків першого рівня.

Формули та рівняння розташовують безпосередньо після тексту, в якому вони згадуються, посередині сторінки. Вище і нижче кожної формули або рівняння повинно бути залишено не менше одного вільного рядка.

Формули і рівняння у пояснювальній записці (за винятком формул і рівнянь, наведених у додатках) слід нумерувати порядковою нумерацією в межах розділу. Номер формули або рівняння складається з номера розділу і порядкового номера формули або рівняння, відокремлених крапкою, наприклад, формула (1.3) – третя формула першого розділу. Номер формули або рівняння зазначають на рівні формули або рівняння в дужках у крайньому правому положенні на рядку. Нумерувати слід лише ті формули, на які є посилання у наступному тексті. Інші нумерувати не рекомендується. Пояснення значень символів і числових коефіцієнтів, що входять до формули чи рівняння, слід наводити безпосередньо під формулою у тій послідовності, в якій вони наведені у формулі чи рівнянні. Пояснення значення кожного символу та числового коефіцієнта слід давати з нового рядка. Перший рядок пояснення починають з абзацу словом «де» без двокрапки.

Переносити формули чи рівняння на наступний рядок допускається тільки на знаках виконуваних операцій, повторюючи знак операції на початку наступного рядка. Коли переносять формули чи рівняння на знакові операції множення, застосовують знак «×».

Загальне правило пунктуації в тексті з формулами таке: формула входить до речення як його рівноправний елемент. Тому в кінці формул і в тексті перед ними розділові знаки ставлять відповідно до правил пунктуації. Двокрапку перед формулою ставлять лише у випадках, передбачених правилами пунктуації: а) у тексті перед формулою є узагальнююче слово; б) цього вимагає побудова тексту, що передує формулі. Розділовими знаками між формулами, котрі йдуть одна за одною і не відокремлені текстом, можуть бути кома або крапка з комою безпосередньо за формулою до її номера. Розділові знаки між формулами при парантезі ставлять всередині парантеза. Після таких громіздких математичних виразів, як визначники і матриці, можна розділові знаки не ставити.

Формули виконують за допомогою вбудованого у MS Word for Windows редактора Equation 3.0. Формули нумерують у круглих дужках праворуч. Формули великого розміру записують у декілька рядків. Перед формулою та

після неї пропускають по одному рядку.

Посилання в тексті на джерела слід зазначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад, «у роботах [1 - 7] ...». Допускається наводити посилання на джерела у виносках, при цьому оформлення посилання має відповідати його бібліографічному опису за переліком посилань із зазначенням номера. При посиланнях на розділи, підрозділи, пункти, підпункти, ілюстрації, таблиці, формули, рівняння, додатки зазначають їх номери. При посиланнях слід писати: «... у розділі 4 ...», «... дивись 2.1 ...», «... за 3.3.4 ...», «... відповідно до 2.3.4.1 ...», «... на рис. 1.3 ...», або «...на рисунку 1.3 ...», «... у таблиці 3.2 ...», «... (див. 3.2) ...», «...за формулою (3.1) ...», «... у рівняннях (1.23) - (1.-25) ...», «... у додатку Б ...».

Текст пояснювальної записки повинний містити: титульний лист; завдання; реферат; зміст; перелік умовних позначень, символів, одиниць, скорочень і термінів; вступ; основний текст; висновки; перелік посилань; додатки.

6. ПОДАННЯ НА ПЕРЕВІРКУ, ПІДГОТОВКА ДО ЗАХИСТУ І ЗАХИСТ КУРСОВОГО ПРОЕКТУ

6.1 Підготовка до захисту курсового проекту

Курсовий проект подається на кафедру за два тижні до її захисту. Проект рецензується (бланк рецензії див. Додаток В). У своїй рецензії викладач зазначає позитивні сторони й недоліки курсового проекту, оцінює ступінь самостійності формулювання основних положень та висновків, наявність елементів творчого пошуку й новизни, величину масиву опрацьованої інформації, дотримання вимог щодо змісту й оформлення проекту, а також робить висновок щодо допуску до захисту з попередньою оцінкою. У разі незадовільної оцінки курсовий проект до захисту не допускається та повертається студенту на доопрацювання. Студент повинен усунути показані в рецензії недоліки і повернути проект для повторного рецензування. Удруге курсовий проект здається з попередньою рецензією.

Недоліки курсових проектів, що часто стають причиною незадовільної оцінки:

- відхід від теми (тема курсового проекту передбачає розкриття одних питань, а фактично розкриваються інші);
- безсистемний виклад матеріалу, повторення одних і тих самих положень;
- логічні помилки, невміння виокремити головне;
- невдале поєднання теорії з фактичним матеріалом;
- переписування матеріалу з друкованих видань;
- недотримання методичних рекомендацій щодо оформлення проекту.

Якщо попередня оцінка позитивна, студент допускається до захисту курсового проекту.

6.2. Захист курсового проекту та критерії оцінки

Якщо проект допущений до захисту, студент повинен ознайомитися з рецензією та підготуватися до захисту. При цьому він має підготувати відповіді на питання, згадані в рецензії, та показати виправлені недоліки, виявлені рецензентом.

Захист курсових проектів здійснюється за встановленим графіком, прилюдно, перед комісією в складі 2-3 викладачів кафедри за участю керівника курсового проекту.

Процедура захисту передбачає стислий виклад студентом головних проблем дослідження та їх вирішення, відповіді на запитання членів комісії. За 10 - 15 хвилин студент повинен коротко викласти мету, зміст та висновки з теми

проведеного дослідження.

У процесі захисту членами комісії оцінюється глибина знань студентом досліджуваної теми, уміння вести дискусію, обґрунтовувати й відстоювати свою точку зору, чітко відповідати на поставлені запитання.

Захист курсового проекту проводиться перед початком сесії.

При оцінці курсового проекту беруться до уваги:

- зміст роботи;
- якість її виконання;
- оформлення проекту згідно з вимогами;
- повноту й точність відповіді на запитання.

Оцінювання курсового проекту здійснюється у два етапи. На основі поданого на кафедру курсового проекту керівник проводить попереднє оцінювання якості її виконання та відповідності вимогам і приймає рішення про допуск (недопуск) виконаного проекту до захисту. Підсумкова оцінка виставляється за результатами захисту курсового проекту перед комісією кафедри з урахуванням попередньої оцінки за курсовий проект. За невиконання курсового проекту чи неподання її на перевірку студенту виставляється незадовільна оцінка. Невиконаним курсовим проектом визнається також у разі її невідповідності встановленим вимогам.

Остаточна оцінка вноситься у відомість та залікову книжку студента і перезахисту не підлягає.

Курсовий проект оцінюється «відмінно», якщо він:

- цілком відповідає вимогам чинних методичних рекомендацій;
- виконаний на високому теоретичному і практичному рівнях;
- містить елементи самостійного дослідження (новизни);
- свідчить про досконале володіння сучасними методами дослідження з використанням комп'ютерних технологій;
- на захисті студент продемонстрував глибоке розуміння проблем теми дослідження, дав вичерпні обґрунтовані відповіді на всі поставлені запитання.

Курсовий проект оцінюється «добре», якщо він:

- цілком відповідає вимогам методичних рекомендацій;
- відповідає не менше трьом вимогам до оцінки «відмінно»;
- на захисті студент показав достатнє розуміння проблем теми курсового проекту і дав майже повні відповіді на поставлені запитання.

Курсовий проект оцінюється «задовільно» за таких умов:

- курсовий проект, в основному, відповідає вимогам;
- теоретичні питання, в основному, розкриті;
- висновки, в основному, правильні, але пропозиції недостатньо

аргументовані;

– на захисті студент правильно охарактеризував основні положення проекту, але не завжди правильно відповідав на поставлені запитання.

Курсовий проект оцінюється «незадовільно» якщо:

- він не відповідає більшій частині поставлених вимог;
- зміст теми не розкрито;
- присутні елементи плагіату;
- проект виконано на основі недіючих нормативних документів;
- на захисті студент не виявив знання основних положень проекту.

У більш узагальненому вигляді рівень виконання курсового проекту оцінюється за такими критеріями:

Критерій	Оцінка у балах (максимальна кількість балів)
Якість викладу матеріалу, пропорційність змісту відповідно до плану теми (повна, часткова, неповна, достатня, недостатня)	20 балів
Наявність у проекті аналізу альтернативних теоретичних поглядів, авторського ставлення, висновків	15 балів
Якісний рівень використання літератури (науково-теоретична, матеріали конференцій, статистичні матеріали, власні розрахунки тощо)	15 балів
Наукова складова	20 балів
Наявність творчості у розкритті змісту теми, вміння користуватися першоджерелами, оформлення схем, графіків, таблиць	20 балів
Загальний вигляд проекту, рівень її оформлення	10 балів
Загальна оцінка у балах	100 балів

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 – 100	A	відмінно
82-89	B	добре
75-81	C	
66-75	D	
60-65	E	задовільно
35-59	FX	незадовільно з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Буч Г., Максимчук Р., Энгл М. Объектно-ориентированный анализ и проектирование с примерами приложений. М.: ООО «ИД Вильямс», 2008. 720 с.
2. Гаврилова Т. А., Хорошевский В. Ф. Об'єктно-орієнтоване програмування. СПб. : Питер, 2000, 384с.
3. Гамма З., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно ориентированного проектирования. Паттерны проектирования. СПб: Питер, 2001. 368 с.
4. Дейт Дж. Введение в объектно-ориентированное программирование. М.: Вильямс, 2005. 1328 с.
5. Карпова Т. ООП. Модели, разработка, реализация: учебник. СПб. : 2001. 302 с.
6. Конноли Т. ООП: проектирование, реализация и сопровождение: учебн. пособ. Теория и практика; пер. с англ. М. : Издательский дом «Вильямс», 2000. 1120 с.
7. Кузин А. В., Левонисова С. В. ООП: учебн. пособие для студентов высш. учебн. М.: Издательский центр «Академия», 2008. 320 с.
8. Павловская Т. А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2003. 461с.
9. Чубук В. В., Чен Р. М., Павленко Л. А. Об'єктно-орієнтоване програмування у питаннях і відповідях : [навч. посібн.]. Х. : Вид. ХНЕУ, 2004. 288 с.
10. Леоненков А. Самоучитель UML 2. СПб. : ВHV, 2007. 576 с.
11. Шаммас Н. К. Основы С++ и объектно-ориентированного программирования. К. : Диалектика, 1996. 448с.
12. Элиенс А. Принципы объектно-ориентированной разработки программ. М.: Издательский дом «Вильямс», 2002. 496 с.

Додаток А

Орієнтовна тематика курсових проектів з дисципліни «Об'єктно-орієнтовне програмування»

1. Інтерпретатор арифметичних виразів.
2. Бібліотека класів для створення текстових інтерфейсів користувача на основі меню.
3. Бібліотека класів для шифрування файлів
4. Об'єкт «комплексні числа». Операції визначаються за загально прийнятим формулами. Передбачити можливість арифметичних операції присвоювання, додавання, множення і перекладу в текстовий рядок поточних значень. Конструктор повинен дозволити створювати об'єкти без та з початковою ініціалізацією.
5. Об'єкт «комплексні числа». Операції визначаються за загально прийнятим формулами. Передбачити можливість операції присвоювання, віднімання, множення і перекладу в текстовий рядок поточних значень. Конструктор повинен до-зволити створювати об'єкти без та з початковою ініціалізацією.
6. Об'єкт «комплексні числа». Операції визначаються за загально прийнятим формулами. Передбачити можливість операції присвоєння, додавання, ділення і перекладу в текстовий рядок поточних значень. Конструктор повинен дозволи-ти створювати об'єкти без та з початковою ініціалізацією.
7. Об'єкт «комплексні числа». Операції визначаються за загально прийнятим фор-мулами. Передбачити можливість операції присвоєння, складання, множення і перекладу в показову форму з можливістю друку на консоль. Конструктор по-винен дозволити створювати об'єкти без та з початковою ініціалізацією.
8. Об'єкт «вектор на площині» заданий у системі декартових координат. Початок вектора розташоване на початку координат. Операції визначаються згідно загально прийнятих формул лінійної (векторної) алгебри. Передбачити можливість операції присвоєння, додавання, скалярного множення і друку координат поточних значень. Конструктор повинен дозволити створювати об'єкти без та з початковою ініціалізацією.
9. Об'єкт «вектор на площині» заданий у системі декартових координат. Початок вектора розташоване на початку координат. Операції визначаються згідно загально прийнятих формул лінійної (векторної) алгебри. Передбачити можливість операції присвоєння, віднімання, скалярного множення і друку координат по-точних значень. Конструктор повинен дозволити створювати об'єкти без та з початковою ініціалізацією.

10. Об'єкт «вектор на площині» заданий у системі декартових координат. Початок вектора розташоване на початку координат. Операції визначаються згідно загально прийнятих формул лінійної (векторної) алгебри. Передбачити можливість операції присвоєння, порівняння модулів, скалярного множення і друку координат поточних значень. Конструктор повинен дозволити створювати об'єкти без та з початковою ініціалізацією.
11. Об'єкт «вектор на площині» заданий у системі декартових координат. Початок вектора розташоване на початку координат. Операції визначаються згідно загально прийнятих формул лінійної (векторної) алгебри. Передбачити можливість операції присвоєння, знаходження кута між векторами, скалярного множення і друку координат поточних значень. Конструктор повинен дозволити створювати об'єкти без та з початковою ініціалізацією.
12. Реалізувати структуру даних стек, яка міститиме наступні методи: додати елемент до стеку, вилучити елемент зі стеку, отримати значення з вершини стеку, розмір стеку, зчитати стек із стандартного потоку, записати стек до потоку стандартного.
13. Реалізувати структуру даних черга, яка міститиме наступні методи: додати елемент до кінця черги, вилучити елемент з початку черги, отримати значення з початку черги, розмір черги, зчитати чергу із стандартного потоку, записати чергу до потоку стандартного.
14. Реалізувати структуру даних двонаправлена черга, яка міститиме наступні методи: додати елемент до кінця та початку черги, вилучити елемент з початку та кінця черги, отримати значення з початку та кінця черги, розмір черги, зчитати чергу із стандартного потоку, записати чергу до потоку стандартного.
15. Реалізувати структуру даних матриця, яка міститиме наступні методи: додати рядок до матриці, вилучити рядок з матриці, отримати та встановити значення елементу матриці, розмір матриці, зчитати матрицю із стандартного потоку, записати матрицю до потоку стандартного.
16. Реалізувати структуру даних вектор, яка міститиме наступні методи: додати елемент до вектора, вилучити елемент з вектора, отримати та встановити значення елементу вектора, розмір вектора, зчитати вектор із стандартного потоку, записати вектор до потоку стандартного.
17. Реалізувати структуру даних список, яка міститиме наступні методи: додати елемент в кінець, початок та довільне місце списку, вилучити елемент з довільного місця списку, отримати та встановити довільне значення елементу списку, розмір списку, зчитати список із стандартного потоку, записати список до по-току стандартного.

18. Реалізувати структуру даних дерево, яка міститиме наступні методи: отримати правий або лівий вузол дерева, отримати сусідній вузол, додати вузол до дере-ва, вилучити вузол з дерева, отримати та встановити довільне значення елементів вузла дерева. Реалізувати метод прямого обходу дерева.
19. Реалізувати структуру даних дерево, яка міститиме наступні методи: отримати правий або лівий вузол дерева, отримати сусідній вузол, додати вузол до дерева, вилучити вузол з дерева, отримати та встановити довільне значення елементів вузла дерева. Реалізувати метод зворотного обходу дерева.
20. Реалізувати структуру даних хеш-таблиця, яка міститиме наступні методи: встановити значення за ключем, перевірити чи наявне значення за ключем, ви-лучити значення за ключем.
21. Реалізація операцій над поліномами довільного степеня
22. Створити бібліотеку класів для роботи з випадковими величинами, реалізувати методи генерації послідовностей випадкових чисел розподілених за: рівномір-ним, нормальним, біноміальним законом розподілу.
23. Створити бібліотеку для методу сортування вставками.
24. Створити бібліотеку для методу швидкого сортування
25. Пошук найкоротшого шляху на площині (розбитій на клітки) між двома клітками.
26. Створити бібліотеку для побудови графіків двовимірних функцій.
27. Створити бібліотеку для побудови векторних полів.
28. Створити бібліотеку для пошуку найкоротшого шляху за алгоритмом Дейкстри.
29. Створити бібліотеку для розв'язування нелінійних рівнянь.
30. Створити інформаційно-пошукову систему: земельний реєстр.

Додаток Б
Зразок оформлення титульної сторінки

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕКОНОМІКИ ТА ПІДПРИЄМНИЦТВА
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ЕКОНОМІЧНОЇ
КІБЕРНЕТИКИ

КУРСОВИЙ ПРОЕКТ

з дисципліни «Об'єктно-орієнтовне програмування»

на тему:

Студент(ка) _____ курсу групи _____
галузі знань _____
спеціальності _____

(прізвище та ініціали студента)

Керівник

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____
Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

м. Вінниця - 20 рік

Додаток В
Зразок бланка рецензії на курсовий проект

РЕЦЕНЗІЯ
на курсовий проект
з дисципліни «Об'єктно-орієнтовне програмування»
студента

_____ (прізвище, ім'я, по батькові)

Тема _____

Обсяг роботи ____ стор., ____ табл., ____ рис., ____ додатків, опрацьованих літературних та інформаційних джерел _____

Позитивні моменти проекту:

Недоліки роботи:

Оформлення роботи

Загальні висновки _____

На доопрацювання _____ «_____» _____ 20 р.

Допущений до захисту _____ «_____» _____ 20 р.

Додаток Г
Зразок завдання курсового проекту

Вінницький національний аграрний університет

Кафедра комп'ютерних наук та економічної кібернетики
Дисципліна Об'єктно-орієнтовне програмування
Спеціальність 122 Комп'ютерні науки та інформаційні технології
Курс 2 **Група** 21КН **Семестр** 3

ЗАВДАННЯ
на курсовий проект

студента Стогнія Костянтина Олександровича
(прізвище, ім'я, по батькові)

Тема курсового проекту (роботи) Бібліотека для методу швидкого сортування

Строк здачі студентом закінченого проекту (роботи) 28.11.2019 р.

Вихідні дані до проекту (роботи) фахова та професійна література, науково-методична література; періодичні видання; Інтернет-джерела

Зміст розрахунково-пояснювальної записки (перелік питань) Метод швидкого сортування. Формулювання вимог до програми. Проектування програми. Кодування програми. Тестування програми. Висновки. Література.

Перелік графічного матеріалу рисунки, таблиці

Дата видачі завдання 11.09.2019 р.

Календарний план

№ пор.	Найменування етапів курсової роботи	Строк виконання роботи
1	<i>Вступ.</i>	<i>25.09.2019 р.</i>
2	<i>Розділ 1</i>	<i>16.10.2019 р.</i>
3	<i>Розділ 2</i>	<i>30.10.2019 р.</i>
4	<i>Розділ 3</i>	<i>13.11.2019 р.</i>
	<i>Висновки. Список використаних джерел</i>	<i>27.11.2019 р.</i>

Студент _____
(підпис)

Керівник _____
(підпис)

Цирульник Сергій Михайлович
(прізвище, ім'я, по батькові)

Додаток Д
Приклади оформлення бібліографічного опису у списку джерел
до курсового проекту

НОВІ ПРАВИЛА ОФОРМЛЕННЯ REFERENCES
з урахуванням ДСТУ 8302:2015, наказу МОН України № 40 від
12.01.2017 р. та міжнародного стилю APA

КНИГИ

Список літератури	References
Однотомні видання	
Один автор	
Битяк Ю. П. Державна служба в Україні: організаційно-правові засади: монографія. Харків: Право, 2005. 304 с.	Bytiak, Yu.P. (2005). Derzhavna sluzhba v Ukraini: orhanizatsiino-pravovi zasady. Kharkiv: Pravo [in Ukrainian].
Грошевий Ю. М. Вибрані праці / упоряд.: О. В. Капліна, В. І. Маринів. Харків: Право, 2011. 656 с.	Hroshevyi, Yu.M. (2011). Vybrani pratsi. Kharkiv: Pravo [in Ukrainian].
Тертишник В. М. Науково-практичний коментар Кримінального процесуального кодексу України: із змінами та допов. на 12 берез. 2016 р. 12-те вид., допов. і переробл. Київ: Правова єдність, 2016. 810 с.	Tertyshnyk, V.M. (2016). Naukovopraktychnyi komentar Kryminalnoho protsesualnoho kodeksu Ukrainy: iz zminamy ta dopov. na 12 berez. 2016 r. Kyiv: Pravova yednist [in Ukrainian].
Johnson L. K. Bombs, bugs, drugs and thugs: intelligence and America's quest for security. New York; London: New York University Press, 2000. 326 p.	Johnson, L.K. (2000). Bombs, bugs, drugs and thugs: intelligence and America's quest for security. New York; London: New York University Press.
Два автори	
Васильєв С. В., Ніколенко Л. М. Доказування та докази у господарському процесі України: монографія. Харків: Еспада, 2004. 192 с.	Vasyliiev, S.V., Nikolenko, L.M. (2004). Dokazuvannia ta dokazy u hospodarskomu protsesi Ukrainy. Kharkiv: Espada [in Ukrainian].
Три автори	
Комаров В. В., Світлична Г. О., Удальцова І. В. Окреме провадження: монографія / за ред. В. В. Комарова. Харків: Право, 2011. 312 с.	Komarov, V.V., Svitlychna, H.O., Udaltsova, I.V. (2011). Okreme provadzhennia. V.V. Komarov (Ed.). Kharkiv: Pravo [in Ukrainian].
Чотири і більше авторів	
Григоренко Є. І., Григоренко Я. О., Козлов В. І. та ін. Колективні політичні права і свободи людини та громадянина в Україні: проблеми теорії та практики: монографія / Харків. нац. ун-т ім. В. Н. Каразіна. Харків, 2013. 352 с.	Hryhorenko, Ye.I., Hryhorenko, Ya.O., Kozlov, V.I. et al. (2013). Kolektyvni politychni prava i svobody liudyny ta hromadianyna v Ukraini: problemy teorii ta praktyky. Kharkiv [in Ukrainian].

Без автора	
Конституція України: наук.-практ. комент. / редкол.: В. Я. Тацій (голова) та ін. 2-е вид., переробл. і допов. Харків: Право, 2012. 1128 с.	Konstytutsiia Ukrainy: nauk.-prakt. koment. V.Ya. Tatsii (Ed.) et al. (2012). Kharkiv: Pravo [in Ukrainian].
Протидія терористичній діяльності: міжнародний досвід і його актуальність для України: матеріали міжнар. наук.-практ. конф. (30 верес. 2016 р., Київ). Київ, 2016. 432 с.	Protydiia terorystychnii diialnosti: mizhnarodnyi dosvid i yoho aktualnist dlia Ukrainy. (2016). Mizhnar. nauk.-prakt. konf. (30 veres. 2016 r., Kyiv) – International Scientific and Practical Conference. Kyiv [in Ukrainian].
Багатотомні видання	
Правова система України: історія, стан та перспективи: у 5 т. / Акад. прав. наук України. Харків: Право, 2009. Т. 2: Конституційні засади правової системи України і проблеми її вдосконалення / за заг. ред. Ю. П. Битяка. 576 с.	Pravova systema Ukrainy: istoriia, stan ta perspektyvy. (Vols. 1–5); Vol. 2: Konstytutsiini zasady pravovoi systemy Ukrainy i problemy yii vdoskonalennia. (2009). Yu.P. Bytiak (Ed.). Kharkiv: Pravo [in Ukrainian].
Кримінальне право України. Загальна частина: підручник: у 2 т. / за ред.: В. В. Сташис, В. Я. Тацій. 4-те вид., переробл. і допов. Харків: Право, 2010. Т. 1. 456 с.	Kryminalne pravo Ukrainy. Zahalna chastyna. V.V. Stashys, V.Ya. Tatsii (Eds.). (2010). (Vols. 1-2; Vol. 1). Kharkiv: Pravo [in Ukrainian].

ІНШІ ВИДАННЯ

Список літератури	References
Дисертації	
Головкін Б. М. Теоретичні та прикладні проблеми детермінації і запобігання корисливій насильницькій злочинності в Україні: дис. ... д-ра юрид. наук. Харків, 2011. 406 с.	Golovkin, B.M. (2011). Teoretychni ta prykladni problemy determinatsii i zapobihannia koryslyvii nasylnytskii zlochynnosti v Ukraini. <i>Doctor's thesis</i> . Kharkiv [in Ukrainian].
Костенко В. О. Економіко-правове забезпечення використання та охорони земель: дис. ... канд. юрид. наук. Харків, 2015. 183 с.	Kostenko, V.O. (2015). Ekonomiko-pravove zabezpechennia vykorystannia ta okhorony zemel. <i>Candidate's thesis</i> . Kharkiv [in Ukrainian].
Автореферати дисертацій	
Наконечний А. Б. Примусове відчуження земельних ділянок за законодавством України: автореф. дис. ... канд. юрид. наук. Харків, 2016. 24 с.	Nakonechny, A.B. (2016). Prymusove vidchuzhennia zemelnykh dilianok za zakonodavstvom Ukrainy. <i>Extended abstract of candidate's thesis</i> . Kharkiv [in Ukrainian].
Стандарти	
ДСТУ ISO 6107-1:2004. Якість води. Словник термінів. Частина 1 (ISO 61071:1996, IDT). [Чинний від 2005-04-01]. Вид. офіц. Київ: Держспоживстандарт України, 2006. 181 с.	DSTU ISO 6107-1:2004. (2006). Yakist vody. Slovnyk terminiv. Chastyna 1 (ISO 61071:1996, IDT). [Chynnyi vid 2005-04-01]. Vyd. ofits. Kyiv: Derzhspozhyvstandart Ukrainy [in Ukrainian].

Архівні документи	
Матеріали Ради Народних комісарів Української Народної Республіки. ЦДАВО України (Центр. держ. архів вищ. органів влади та упр. України). Ф. 1061. Оп. 1. Спр. 8–12. Копія; Ф. 1063. Оп. 3. Спр. 1–3.	Materialy Rady Narodnykh komisariv Ukrainskoi Narodnoi Respubliki. TsDAVO Ukrainy (Tsent. derzh. arkhiv vyshch. orhaniv vlady ta upr. Ukrainy). F. 1061. Op. 1. Spr. 8–12. Kopia; F. 1063. Op. 3. Spr. 1–3 [in Ukrainian].
Бібліографічні покажчики	
Володимир Володимирович Сташис: (до 85-річчя від дня народж. та 60-річчя наук.пед. і громад. діяльності); упоряд.: В. І. Борисов, В. І. Тютюгін, Л. М. Демидова. Харків: Право, 2010. 108 с.	Volodymyr Volodymyrovych Stashys: (do 85richchia vid dnia narodzh. ta 60-richchia nauk.ped. i hromad. diialnosti). (2010). V.I. Borysov, V.I. Tiutiuhin, L.M. Demydova (Ed.). Kharkiv: Pravo [in Ukrainian].

ЧАСТИНА ВИДАННЯ

Список літератури	References
Розділ книги	
Борисова В. И. Право частной собственности в системе социальноэкономических прав и свобод граждан и пути его реализации. Харьковская цивилистическая школа: право собственности: монография / под ред. И. В. Спасибо-Фатеевой. Харьков: Право, 2012. Разд. 3, гл. 1. С. 87–99.	Borysova, V.Y. (2012). Pravo chastnoi sobstvennosti v systeme sotsyalnoekonomycheskykh prav y svobod hrazhdan y puty eho realizatsyy. <i>Kharkovskaia tsyvylystycheskaia shkola: pravo sobstvennosti – Kharkov Civil School: Ownership</i> . Y.V. Spasybo-Fateeva (Ed.). Kharkov: Pravo, part. 3, ch. 1, 87–99 [in Russian].
Матеріали конференцій, круглих столів	
Боднар Т. В. Договір про закупівлю: особливості укладання і забезпечення. <i>Актуальні проблеми приватного права: договір як правова форма регулювання приватних відносин</i> : матеріали наук.практ. конф., присвяч. 95-й річниці з дня народж. В. П. Маслова (Харків, 17 лют. 2017 р.). Харків: Право, 2017. С. 7–9.	Bodnar, T.V. (2017). Dohovir pro zakupivliu: osoblyvosti ukladannia i zabezpechennia. <i>Aktualni problemy pryvatnoho prava: dohovir yak pravova forma rehuliuvannia pryvatnykh vidnosyn</i> : proceedings of the Scientific and Practical Conference. Kharkiv: Pravo, 7–9 [in Ukrainian].
Стаття з продовжаного видання	
Гетьман А. П., Лозо В. І. Державно-правові проблеми подолання екологічної кризи в епоху глобалізації. <i>Проблеми законності</i> . Харків. 2013. Вип. 123. С. 65–77.	Getman, A.P., Lozo, V.I. (2013). Derzhavnopravovi problemy podolannia ekolohichnoi kryzy v epokhu hlobalizatsii [Some state-legal problems of overcoming environmental crisis in globalisation era]. <i>Problemy zakonnosti – Problems of Legality, issue 123, 65–77</i> [in Ukrainian].
Стаття з періодичного видання (журнал, газета)	
Петришин О., Серьогіна С. Змішана республіканська форма державного правління: питання теорії та практики. <i>Право України</i> . 2009. № 10. С. 57–60.	Petryshyn, O., Serohina, S. (2009). Zmishana respublikanska forma derzhavnoho pravlinnia: pytannia teorii ta praktyky. <i>Pravo Ukrainy – Law of Ukraine, 10, 57–60</i> [in Ukrainian].

Таций В. Я., Тютюгін В. І., Пономаренко Ю. А. Виклики сучасності і кримінальне право. <i>Голос України</i> . 2016. 29 січ. (№ 16). С. 6–7.	Tatsii, V.Ya., Tiutiuhin, V.I., Ponomarenko, Yu.A. (2016). Vyklyky suchasnosti i kryminalne pravo. <i>Holos Ukrainy – Voice of Ukraine</i> , 29 sich. (No. 16), 6–7 [in Ukrainian].
Benjamin A. C. The ethics of scholarship: A discussion of problems that arise in its application. <i>Journal of Higher Education</i> . 1960. Vol. 31, No. 9. P. 471–480.	Benjamin, A.C. (1960). The ethics of scholarship: A discussion of problems that arise in its application. <i>Journal of Higher Education</i> . Vol. 31, 9, 471–480.

ЕЛЕКТРОННІ РЕСУРСИ

Список літератури	References
Гетьман Є. А. Підзаконні нормативноправові акти органів виконавчої влади України та іноземних держав: порівняльна характеристика. <i>Теорія і практика правознавства</i> : електрон. наук. фахове вид. 2016. Вип. 1 (9). URL: http://tlaw.nlu.edu.ua/article/view/66302 (дата звернення: 17.06.2016).	Hetman, Ye.A. (2016). Pidzakonni normatyvnopravovi akty orhaniv vykonavchoi vldy Ukrainy ta inozemnykh derzhav: porivnialna kharakterystyka. <i>Teoriia i praktyka pravoznavstva – Theory and Practice of Jurisprudence</i> , issue 1 (9). URL: http://tlaw.nlu.edu.ua/article/view/66302 [in Ukrainian].
Карнаух Б. П. Тлумачення договору: короткий нарис із наднаціональної і транснаціональної точок зору. <i>Проблеми законності</i> . 2016. Вип. 135. С. 39–51. doi: http://dx.doi.org/10.21564/2414-990x.135.83852 .	Karnaukh, B.P. (2016). Tlumachennia dohovoru: korotkyi narys iz nadnatsionalnoi i transnatsionalnoi tochok zoru [Contract Interpretation: Brief Overview from the Standpoint of International and Trans-National Instruments]. <i>Problemy zakonnosti – Problems of Legality</i> , issue 135, 39–51. doi: http://dx.doi.org/10.21564/2414-990x.135.83852 [in Ukrainian].
Аналіз стану здійснення судочинства в 2015 році (за даними судової статистики). URL: http://www.scourt.gov.ua/clients/vsu/vsu.nsf/(documents)/D7F9F72E78DA88ECC2257F730036F282 (дата звернення: 17.03.2017).	Analiz stanu zdiisnennia sudochynstva v 2015 rotsi (za danymy sudovoi statystyky. URL: http://www.scourt.gov.ua/clients/vsu/vsu.nsf/(documents)/D7F9F72E78DA88ECC2257F730036F282 [in Ukrainian].

ЗАКОНОДАВЧІ ТА НОРМАТИВНІ ДОКУМЕНТИ

Список літератури	References
Кримінальний кодекс України: Закон України від 05.04.2001 р. № 2341-III. <i>Відомості Верховної Ради України</i> . 2001. № 25–26. Ст. 131.	Kryminalnyi kodeks Ukrainy: Zakon Ukrainy vid 05.04.2001 r. № 2341-III. (2001). <i>Vidomosti Verkhovnoi Rady Ukrainy</i> , 25–26, art. 131.
Про порядок класифікації надзвичайних ситуацій: Постанова Кабінету Міністрів України від 15.06.1998 р. № 1099. <i>Офіційний вісник України</i> . 1998. № 28. Ст. 1062.	Pro poriadok klasyfikatsii nadzvychainykh sytuatsii: Postanova Kabinetu Ministriv Ukrainy vid 15.06.1998 r. № 1099. (1998). <i>Ofitsijnyj visnyk Ukrainy – Official Gazette of Ukraine</i> , 28, art. 1062.

<p>Про правовий режим воєнного стану: Закон України від 12.05.2015 р. № 389-VIII. <i>Голос України</i>. 2015. 10 черв. (№ 101). С. 12.</p>	<p>Pro pravovyi rezhym voiennoho stanu: Zakon Ukrainy vid 12.05.2015 r. № 389-VIII. (2015). <i>Holos Ukrainy – Voice of Ukraine</i>, 101, 12.</p>
<p>Інструкція про призначення та проведення судових експертиз та експертних досліджень: затв. наказом М-ва юстиції України від 08.10.1998 р. № 53/5 (у ред. наказу від 26.12.2012 р. № 1950/5). <i>Офіційний вісник України</i>. 2013. № 3. Ст. 91.</p>	<p>Instruktsiia pro pryznachennia ta provedennia sudovykh ekspertyz ta ekspertnykh doslidzhen: zatv. nakazom M-va yustytsii Ukrainy vid 08.10.1998 r. № 53/5 (u red. nakazu vid 26.12.2012 r. № 1950/5). (2013). <i>Ofitsijnyj visnyk Ukrainy – Official Gazette of Ukraine</i>, 3, art. 91.</p>
<p>Про запобігання корупції: Закон України від 14.10.2014 р. № 1700-VII. Дата оновлення: 12.03.2017. URL: http://zakon0.rada.gov.ua/laws/show/1700-18/page (дата звернення: 17.03.2017).</p>	<p>Pro zapobihannia koruptsii: Zakon Ukrainy vid 14.10.2014 r. № 1700-VII. Data onovlennia: 12.03.2017. URL: http://zakon0.rada.gov.ua/laws/show/1700-18/page.</p>
<p>Ухвала апеляційного суду Полтавської області від 27 серпня 2014 р., судова справа № 551/818/14-к. URL: http://www.reyestr.court.gov.ua/Review/41131992 (дата звернення: 17.06.2016).</p>	<p>Ukhvala apeliatsiinoho sudu Poltavskoi oblasti vid 27 serpnia 2014 r., sudova sprava № 551/818/14-k. URL: http://www.reyestr.court.gov.ua/Review/41131992.</p>