

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені ВАДИМА ГЕТЬМАНА»

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

ЗАТВЕРДЖУЮ:

Проректор з науково-педагогічної роботи

_____  А.М. Колот

«28» 10 2021 р.

МЕТОДИЧНІ ВКАЗІВКИ
з написання та захисту курсової роботи
з навчальної дисципліни
Системне програмування

рівень вищої освіти	перший (бакалаврський)
галузь знань	12 Інформаційні технології
спеціальність	122 Комп'ютерні науки
освітньо-професійна програма	Комп'ютерні науки
тип навчальної дисципліни	обов'язкова

Начальник навчально-методичного відділу

 Д.М. Котенок

ПОГОДЖЕНО:

В.о. завідувача кафедри  Б.О. Тішков

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені ВАДИМА ГЕТЬМАНА»

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

ЗАТВЕРДЖУЮ:

Проректор з науково-педагогічної роботи

_____ А.М. Колот

«___» _____ 20__ р.

МЕТОДИЧНІ ВКАЗІВКИ
з написання та захисту курсової роботи
з навчальної дисципліни
Системне програмування

рівень вищої освіти	перший (бакалаврський)
галузь знань	12 Інформаційні технології
спеціальність	122 Комп'ютерні науки
освітньо-професійна програма	Комп'ютерні науки
тип навчальної дисципліни	обов'язкова

Начальник навчально-методичного відділу
_____ Д.М. Котенок

ПОГОДЖЕНО:

В.о. завідувача кафедри _____ Б.О. Тішков

Розробник:

Іванченко Г.Ф., к.т.н., доцент, професор кафедри, кафедра інформаційних систем в економіці, ivan734@i.ua

Форма навчання —	очна (денна)
Семестр —	2
Форма підсумкового контролю —	захист курсової роботи
Мова(и) викладання	українська мова

Форма навчання —	заочна
Семестр —	4
Форма підсумкового контролю —	захист курсової роботи
Мова(и) викладання	українська мова

Форма навчання —	дистанційна
Семестр —	2
Форма підсумкового контролю —	захист курсової роботи
Мова(и) викладання	українська мова

ЗМІСТ

ВСТУП.....	4
1. ОРГАНІЗАЦІЯ ПІДГОТОВКИ КУРСОВОЇ РОБОТИ.....	6
1.1. Загальні положення	6
1.2. Вибір теми роботи	8
1.3. Тематика курсових робіт.....	8
1.4. Етапи виконання роботи	8
1.5. Графік виконання роботи.....	10
2. СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	11
2.1. Титульний аркуш.....	11
2.2. Зміст	11
2.3. Перелік умовних позначень	11
2.4. Вступ	12
2.5. Основна частина	12
2.6. Перший розділ	12
2.7. Другий розділ	13
2.8. Третій розділ.....	13
2.9. Висновки	14
2.10. Додатки	14
2.11. Список використаної літератури	14
3. ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	15
3.1. Загальні вимоги до оформлення роботи	15
3.2. Параметри сторінок.....	15
3.3. Оформлення тексту	15
3.4. Зміст	15
3.5. Заголовки	15
3.6. Таблиці.....	16
3.7. Ілюстрації.....	17
3.8. Додатки	17
3.9. Поради щодо оформлення тексту.....	18
4. ЗАХИСТ РОБОТИ	19
4.1. Документи, які мають бути підготовлені перед захистом	19
4.2. Процедура захисту	19
4.3. Критерії оцінювання	20
5. РЕКОМЕНДОВАНІ ІНФОРМАЦІЙНІ ДЖЕРЕЛА.....	23
5.1. Основна література.....	23
5.2 . Додаткова література	23
5.3. Дистанційні курси та інформаційні ресурси.....	24
ДОДАТКИ	25

ВСТУП

Методичні вказівки з написання та захисту курсової роботи з навчальної дисципліни «Системне програмування» розроблені відповідно до «Положення про порядок оцінювання результатів навчання здобувачів вищої освіти в Державному вищому навчальному закладі «Київський національний економічний університет імені Вадима Гетьмана», затвердженого Вченою радою Університету 27.05.2021 р. (протокол № 10) та введеного в дію наказом ректора від 27.05.2021 р. №305.

Анотація навчальної дисципліни. Курсова робота з навчальної дисципліни «Системне програмування» – є одним з видів науково-дослідної роботи, виконується відповідно до навчального плану освітньо-професійної програми «Комп'ютерні науки» з метою поглиблення та узагальнення знань, отриманих здобувачами вищої освіти за час навчання, та їх практичного застосування до комплексного вирішення конкретного фахового завдання. Курсова робота з навчальної дисципліни «Системне програмування» є заключним етапом вивчення навчальної дисципліни «Системне програмування» і забезпечує закріплення, поглиблення, узагальнення теоретичних знань та практичних навичок, які отримують здобувачі вищої освіти під час вивчення навчальної дисципліни «Системне програмування». До основних задач курсової роботи з навчальної дисципліни «Системне програмування» входить: розвиток у здобувачів навичок науково-дослідної і проектно-конструкторської роботи в області розробки системних програмних засобів. Робота виконується з використанням сучасних інструментальних систем, придбання навичок самостійної роботи над науково-технічною і довідковою літературою за фахом, стандартами, різного виду нормативною документацією й іншими посібниками; самостійним вибором найбільш доцільних і технічно зроблених рішень задач за розглянутим курсом. Під час виконання курсової роботи здобувач повинний одержати навички, що будуть необхідні йому при дипломному проектуванні й у подальшій інженерній діяльності.

Методичні вказівки з написання та захисту курсової роботи з навчальної дисципліни «Системне програмування» є однією із складових комплексної підготовки фахівців галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки».

Міждисциплінарні зв'язки. Для вивчення дисципліни необхідні базові знання з таких дисциплін, як: «Системне програмування», «Програмування та алгоритмічні мови», «Операційні системи», «Математична логіка та теорія алгоритмів».

Мета вивчення навчальної дисципліни полягає в отриманні навичок складання системних програм у відповідності до вимог завдання або технічного завдання і підготовки комплекту документів на програми для її замовника або користувача. Вивчення основних вимог до базових документів системних програм та одержання навичок оформлення таких документів на прикладах реалізації системних програм. Метою навчальної дисципліни також є

формування у здобувачів вищої освіти низки компетентностей, а саме: здатність розуміти принципи керування процесором на програмному рівні; здатність розуміти послідовність дій під час розробки програм на машинній мові; здатність аналізувати структуру програм на машинній мові; здатність розробляти програми на мові Assembler; здатність аналізувати процеси, які здійснюються під час компіляції та лінковки програм; здатність використовувати можливості макроасемблерів, у програмуванні; здатність аналізувати використання програмами ресурсів обчислювальної системи, зокрема пам'яті.

Завдання навчальної дисципліни. Ключовою навчальною ціллю опанування дисципліни є формування у здобувачів вищої освіти визначеного набору компетентностей для підготовки до професійної діяльності. Після засвоєння навчальної дисципліни здобувачі мають продемонструвати такі результати навчання: структури і складові системного програмного забезпечення; структури та формати команд на машинній мові; структури програм на мові Assembler; режими процесора x86-64; основні формати зберігання даних; директиви компілятора Assembler; взаємодію системних програм під час їх виконання; розв'язання задач аналізу та синтезу засобів, характерних для спеціальності 122 «Комп'ютерні науки».

Предмет вивчення навчальної дисципліни: системне програмування, методологія та інструментарій розробки програм.

1. ОРГАНІЗАЦІЯ ПІДГОТОВКИ КУРСОВОЇ РОБОТИ

1.1. Загальні положення

Курсова робота — це індивідуальне завдання здобувачу з дисципліни «Системне програмування», в якому комплексно використовуються знання та вміння, набуті в процесі вивчення дисципліни.

Метою курсової роботи є вивчення здобувачами взаємозв'язку між архітектурою ЕОМ та системними програмами, надання здобувачам базових знань з програмування на платформі MASM32, проектування та розробки системних програм, підвищення та закріплення набутих під час вивчення дисципліни «Системне програмування» теоретичних знань і практичних вмінь здобувачів.

Вміння програмувати відіграє важливу роль не тільки при створенні програмних продуктів, а також при роботі і налагодженні програмного забезпечення. Ефективність даної роботи залежить від вмінь відповідальної особи чітко визначити поставлену проблему та знайти оптимальні шляхи її розв'язання.

«Системне програмування» – це навчальна дисципліна, що розкриває не лише методологію створення системних програм для комп'ютерних систем, а також розкриває основні труднощі, які виникають під час створення подібних програм, та шляхи їх вирішення.

Метою також є розкриття сучасних наукових концепцій, понять, методів та технологій проектування та реалізації системного програмування на підставі засвоєння алгоритмів, що покладені в основу операційних систем, алгоритмів системних служб, алгоритмів обробки інформації різних типів даних, вивчення принципів реалізації системного програмного забезпечення операційних середовищ та систем з використанням сучасних технологій програмування.

Завдання курсової роботи:

- закріплення вмінь самостійно спроектувати загальний алгоритм роботи складної системи, провести необхідні обґрунтування для розробки системних програмних засобів, підібрати оптимальні алгоритми для реалізації окремих складових, скористатись для цього сучасними джерелами інформації, у тому числі Інтернет-ресурсами;
- закріплення вмінь самостійно вибрати середовище розробки;
- закріплення вмінь самостійно реалізувати побудовані алгоритми програмно, довести роботу до логічного кінця, проаналізувати отримані результати;
- ознайомлення з основними етапами розробки закінченого програмного продукту та основними вимогами до оформлення текстового, програмного та ілюстративного матеріалу звіту до нього.
- ознайомленні з принципами побудови системних програм, операційних середовищ та систем;
- вивчення основ програмування;

- засвоєння основних концепцій системного програмування;
- вивчення основних компонентів та особливостей системного програмного забезпечення;
- засвоєння основних засобів та методів конструювання ефективних системних програм;
- набуття практичних навичок програмування на платформі Masm32;
- набуття практичних навичок в програмуванні фрагментів системних програм;
- вивчення і порівняльний аналіз побудови та системних служб сучасних операційних систем;
- вивчення і реалізація основних алгоритмів, покладених в основу операційних середовищ і систем.

Дана курсова робота – це перша самостійна навчально-виробнича розробка здобувачів вищої освіти першого курсу, яка має бути виконана й оформлена відповідно до сучасних технологій дослідження, практичної реалізації та вимог оформлення супровідної документації.

Знання та вміння, здобуті в результаті виконання курсової роботи, в подальшому застосовуватимуться здобувачем при виконання наступних курсових, індивідуальних навчально-дослідних та навчально- виробничих завдань, наукових досліджень та кваліфікаційного бакалаврського проекту.

В результаті виконання курсової роботи здобувач повинен набути такі результати навчання:

1. Знання:

- основні принципи побудови операційних середовищ та систем, платформи Masm32;
- принципи побудови спеціалізованих апаратно-програмних обчислювальних комплексів збирання, обробки та передавання даних;
- алгоритми, покладені в основу операційних середовищ та систем;
- основні компоненти платформи Masm32;
- можливості мови програмування Masm32 при створенні програм;

2. Уміння:

- використовувати основні засоби та методи програмування при конструюванні ефективних систем програмного забезпечення;
- створювати консольні програми у середовищі Masm32;
- розробляти програми мовою програмування Masm32;
- тестувати виконання програми Masm32;
- створювати консольні програми;
- розробляти графічний інтерфейс програми;
- використовувати структури бібліотеки базових класів Masm32 при створенні програм;
- використовувати набуті теоретичні знання з мови програмування Masm32 при створенні системних програм.

1.2. Вибір теми роботи

Здобувачі самостійно вибирають тему роботи із запропонованих або можуть запропонувати свою власну після обговорення її з керівником курсу (лектором), виходячи із власних схильностей, інтересів та знань.

Тематика (перелік можливих тем) курсових робіт затверджується рішенням кафедри і оприлюднюється не пізніше як за 3 місяці до планового терміну їх захисту.

Затвердження тем курсових робіт відбувається на підставі оформлення (разом з керівником) листа Завдання, який подається на кафедру не пізніше як за три місяці до терміну захисту роботи. У разі, якщо здобувач не оформив в означений термін лист Завдання без поважної причини, тема роботи закріплюється на розсуд кафедри.

Не допускається виконання курсових робіт на однакову тему різними здобувачами, тому всі теми курсових робіт погоджуються з керівником курсу.

У випадку виникнення потреби змінити тему курсової роботи здобувач подає на кафедру відповідну мотивовану заяву з візою наукового керівника на ім'я завідувача кафедри. Рішення приймається не пізніше ніж за місяць до планового терміну захисту курсових робіт.

1.3. Тематика курсових робіт

Тематика курсових робіт може бути розділена на окремі групи, кожна із яких має свої особливості при реалізації і орієнтована на застосування спеціальних знань з інших дисциплін, що викладаються або вже були викладені на час виконання курсової роботи.

Нижче надано загальний перелік тем курсових робіт по групам і визначено їх особливості: Етапи створення програми мовою асемблера. Архітектура мікропроцесора INTEL. Основні директиви та операнди асемблера. Операції пересилання даних. Основні арифметичні операції. Команди передання керування. Процедури. Динамічні бібліотеки. Файли. Арифметичний співпроцесор. Макровизначення. Логічні конструкції високого рівня. Цей список може бути розширений за узгодженням з керівником курсу.

1.4. Етапи виконання роботи

Кожний здобувач отримує своє власне індивідуальне завдання, яке повинен виконати протягом семестру.

Виконання здобувачем курсової роботи з дисципліни «Системне програмування» включає в себе такі послідовні етапи:

1. Вибір теми роботи.

2. Обговорення з керівником і оформлення бланка завдання з підписами керівника і здобувача – як закріплення теми індивідуального завдання.

3. Аналіз постановки задачі і вибір методів з використанням відповідних літературних джерел та інтернет-ресурсів.

4. Побудова алгоритму розв'язку задачі. Рекомендований підхід при побудові алгоритму – спадний для побудови загального алгоритму роботи програми як складної структурної системи. Окремо потрібно пропрацювати алгоритми роботи окремих складових системи – підсистем. Алгоритми можуть бути оригінальні – авторські, можна скористатись стандартними класичними алгоритмами. Коректність обраного алгоритму має бути доведена. При побудові загального алгоритму потрібно враховувати можливість легко його удосконалити для збільшення або покращення функціональності.

5. Проектування інтерфейсу. Інтерфейс має бути дружнім до того користувача, на якого розрахований даний програмний продукт. Він має бути добре прокоментований та інтуїтивно зрозумілий для нього. Для будь-якого програмного продукту має бути передбачене основне меню та декілька підменю. В основному меню передбачити пункти допомоги, інформації про розробника, пункт для виходу із програми та інші – відповідно темі.

6. Програмна реалізація. Програма повинна бути реалізована як консольний додаток або як WINDOWS-додаток (з дозволу керівника), представляти собою проект, зібраний з декількох модулів, у тому числі авторських, поданих з використанням заголовкових файлів. Функції, що входять до складу одного файлу, мають бути логічно пов'язані. Окрему функцію бажано будувати таким чином, щоб вона відповідала за якусь одну дію. Користуватися глобальними даними не бажано, оскільки це збільшує контекстну залежність даної функції.

Тестування та налагодження програми з OLLYDBG і аналіз результатів. Тестування програми варто проводити як на стандартних даних (які апріорі дають очікувані результати), так і на нестандартних даних для забезпечення коректної роботи програми на "крайніх" значеннях (наприклад, усі нулі або свідомо завелике число). Програма повинна бути захищена і забезпечувати стабільне виконання, тобто не повинна реагувати на натиснення будь-яких клавіш, крім допустимих на даному етапі виконання. На завершення слід проаналізувати порядок розміщення даних у пам'яті комп'ютера та їх відповідність до адрес. Порядок розміщення даних у пам'яті та стан налагоджувача OllyDbg з програмою з лістингу Додаток Д.

7. Демонстрація програми.

8. Оформлення пояснювальної записки.

9. Здача курсової роботи науковому керівнику на перевірку.

10. Захист курсової роботи.

Університетська освіта передбачає оволодіння навичками самостійно ставити і розв'язувати завдання, самостійно працювати з літературою та оволодівати потрібним інструментарієм для реалізації конкретної задачі. Тому більшу частину роботи здобувач повинен виконувати самостійно, але не втрачати при цьому зв'язку з керівником, головне завдання якого – спрямувати студента на розв'язок конкретних питань з обраної теми, допомогти у пошуку шляхів та методів вирішення проблеми.

Обов'язковими компонентами керівництва виконанням здобувачами курсових робіт є:

- вступна настанова (групова співбесіда) тривалістю до однієї пари з усіма студентами, які виконують курсову роботу під керівництвом цього викладача;
- щотижнево на консультаціях та лабораторних заняттях студент повинен згідно з графіком звітуватись про виконану роботу, і, в разі потреби, отримати консультацію щодо подальшого виконання курсової роботи.

1.5. Графік виконання роботи

Графік виконання курсової роботи представлений у таблиці 1 і є обов'язковим для виконання усіма здобувачами.

Таблиця 1 – Графік виконання курсової роботи з дисципліни «Системне програмування»

№ п/п	Назва етапів курсової роботи	Строк виконання етапів роботи (порядкові тижні семестру)
1.	Вибір теми курсової роботи. Опрацювання відповідної літератури. Оформлення бланка завдання.	1-й, 2-й тижні
2.	Аналіз постановки задачі.	3-й тиждень
3.	Вибір та дослідження методів, вибір відповідних структур даних, розробка алгоритму. Перше узгодження з керівником.	4-й тиждень
4.	Проектування інтерфейсу.	5-й тиждень
5.	Друге узгодження з керівником.	6-й тиждень
6.	Програмна реалізація.	7, 8, 9-й тижні
7.	Демонстрація першого варіанту. Третє узгодження з керівником.	10-й тиждень
8.	Заключне тестування програми.	11-й тиждень
9.	Аналіз результатів. Оформлення звіту.	12, 13-й тиждів
10.	Захист та демонстрація курсової роботи.	14, 15-й тижні

2. СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка до курсової роботи складається з таких структурних елементів:

- титульний аркуш;
- заповнений бланк завдання;
- зміст;
- перелік умовних позначень (за необхідності);
- вступ;
- основна частина (розділи роботи);
- висновки;
- список використаної літератури;
- додатки.

2.1. Титульний аркуш

Титульний аркуш оформлюється виключно згідно наведеного у додатку А зразку.

Тема роботи повинна зазначатись ідентично темі, затвердженій кафедрою. У разі невідповідності робота до захисту не приймається. Прізвище, ім'я та по-батькові здобувача та керівника роботи повинні зазначатись повністю.

Бланк завдання складається з двох сторінок і оформлюється на одному аркуші з двох сторін. Зразок надано у додатку Б.

Поля, заповнені у зразку, можуть бути надруковані. Поля, що передбачають індивідуальну інформацію, потрібно заповнити разом з керівником і поставити свій підпис, а керівник має поставити свій підпис. Заповнити бланк завдання необхідно відповідно графіку виконання роботи.

2.2. Зміст

Зміст пояснювальної записки до курсової роботи повинен послідовно містити назви всіх структурних елементів роботи 1-го та 2-го рівнів (окрім титульного аркуша та самого змісту) та посилання на номери сторінок, на яких починається даний структурний елемент.

Візуально зміст роботи повинен відображати ієрархію структурних елементів роботи (перелік умовних позначень, вступ, розділи, висновки, додатки – елементи 1 рівня; підрозділи — елементи 2 рівня).

На початку по центру розміщується текст "ЗМІСТ" (без лапок), оформлений напівжирним шрифтом. Приклад оформлення змісту пояснювальної записки надано в Додатку Г.

2.3. Перелік умовних позначень

Перелік умовних позначень є обов'язковим елементом роботи. Він складається у випадку, коли робота містить маловідомі скорочення,

аббревіатури, символи, специфічні терміни.

Перелік друкується двома колонками, в яких ліворуч за абеткою наводять позначення чи терміни, праворуч — їх детальне розшифрування (тлумачення).

Якщо в роботі певний термін, скорочення чи позначення повторюється менше трьох разів, його у перелік не включають, а його розшифрування наводять у тексті при першому згадуванні.

2.4. Вступ

Вступ до курсової роботи повинен містити такі обов'язкові елементи (у такому ж порядку):

- актуальність: чому було важливо та актуально обрати саме таку тему;
- мета ("розробити та реалізувати закінчений програмний продукт для ...");
- завдання ("на основі аналізу літературних джерел [визначити основні риси майбутньої розробки, оцінити існуючі методи та алгоритми реалізації її складових], складено структурний алгоритм роботи майбутньої програми ... та реалізовано його у вигляді програмного продукту ..., що складається з ... модулів, у тому числі стандартних ..., з використанням процедурного програмування");
- практичне значення одержаних результатів (можливо, місце застосування програмного продукту або його частини);
- апробація результатів дослідження (необов'язковий елемент; можливо, результати застосування програмного продукту або його частини);
- використане програмне забезпечення ("При виконанні роботи було використане таке програмне забезпечення: стандартне середовище розробки ..., операційна система ...; веб-браузер ... для роботи з веб-сайтами ...; текстовий редактор ... для підготовки та оформлення курсової роботи");
- структура роботи ("Робота складається зі вступу, трьох розділів, висновків, додатків та списку використаних джерел").

Кожен елемент вступу повинен починатись з нового абзацу. Назви елементів повинні виділятися напівжирним шрифтом.

Загальний обсяг вступу не повинен перевищувати 2 сторінок.

2.5. Основна частина

Основна частина роботи складається з трьох послідовних розділів, які присвячені наступним питанням.

2.6. Перший розділ

Перший розділ має назву "ПОСТАНОВКА ЗАДАЧІ". Містить два підрозділи.

Перший підрозділ присвячений огляду існуючих підходів до розв'язання поставленої задачі, аналізу існуючих алгоритмів, варіантів візуалізації (для ігрових програм), тощо.

Другий присвячений обґрунтуванню вибраного варіанту реалізації і детальному опису постановки задачі. Постановка задачі може бути сформульована математично або на змістовному рівні (умови гри для ігрових програм, перелік методів і функцій, що треба реалізувати для інформаційних систем, тощо). Крім того, описуються вимоги до програмних та технічних засобів.

2.7. Другий розділ

Другий розділ має назву "РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ" і включає два підрозділи.

Перший підрозділ має назву "Метод розв'язку задачі" і містить виклад рішення, що пропонується. Залежно від теми роботи, цей підрозділ може включати опис необхідного математичного апарату, фізичного явища, обґрунтування вибору обраного алгоритму розв'язку, опис основних структур даних. У цьому підрозділі можуть бути використані такі елементи як формули, рисунки, таблиці і графіки.

Другий підрозділ має назву "Алгоритм розв'язку задачі", містить опис загального алгоритму роботи всієї програми. Може містити описи одного або декількох самостійно розроблених оригінальних або модифікованих стандартних алгоритмів, що реалізують окремі частини загального алгоритму.

У цьому підрозділі обов'язково має бути присутній рисунок зі структурною блок-схемою узагальненого алгоритму роботи програми (1-2 сторінки), у якій виділяються головні функціональні блоки програми та відображається зв'язок між ними. За необхідності, можуть бути представлені також блок-схеми описаних вище оригінальних алгоритмів [9].

2.8. Третій розділ

Третій розділ "ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ" присвячений реалізації програмного продукту. Містить опис основної програми і користувацьких програмних модулів у порядку їх оголошення. Складається з таких підрозділів: "Опис головних структур і змінних програми", "Опис головних функцій програми", "Опис інтерфейсу" та "Результати роботи програмного продукту".

Два перших підрозділи можуть бути побудовані у вигляді тексту або таблиць, що містять відповідні описи, згруповані за належністю до певного заголовкового файлу. Обов'язково потрібно надавати опис тих структур та змінних, що винесені у заголовкові файли та таких, розуміння призначення яких може вплинути на розуміння функціонування важливої частини або програми загалом. Опис даних містить тип та семантику.

Обов'язково потрібно надавати опис усіх розроблених функцій, незалежно від розміру та призначення. Опис кожної функції містить її прототип та призначення, а також семантику кожного параметра під час виклику.

Приклади відповідних таблиць представлені у Додатку В.

Підрозділ "Опис інтерфейсу" представляє собою фактично керівництво користувача даного програмного продукту і містить описи усіх реалізованих меню та підменю, ілюстровані рисунками з відповідними екранними формами (можуть бути чорно-білі).

Підрозділ "Результати роботи програмного продукту" містить інформацію про результати тестування програми та порівняльну характеристику з існуючими продуктами такого плану, виявлені недоліки та рекомендації щодо використання. Може містити ілюстрації певних ігрових ситуацій, станів інформаційно-пошукової системи або інших результатів роботи програми.

2.9. Висновки

Висновки та пропозиції є стислим викладенням підсумків дослідження. У даному випадку – висновки по розробленому програмному продукту: що і як вдалось реалізувати, переваги та недоліки розробки, можливі шляхи подальшого покращення. Обсяг висновків становить 1–2 сторінки.

2.10. Додатки

Додаток повинен містити текст основної програми та тексти підключених модулів власної розробки (як заголовкові файли, так і файли реалізації) у порядку їх оголошення. Тексти функцій можуть містити стислі коментарі щодо їх призначення.

2.11. Список використаної літератури

Список використаної літератури повинен містити перелік в алфавітному порядку усіх літературних джерел, які були використані при написанні курсової роботи. Джерела оформлюються згідно вимог стандарту ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання» [8].

3. ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка (ПЗ) оформлюється згідно ДСТУ 3008-95 «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення».

3.1. Загальні вимоги до оформлення роботи

Пояснювальна записка друкується у текстовому редакторі на аркушах білого паперу формату А4 лише з одного боку аркушу.

3.2. Параметри сторінок

Ліве поле — 3 см, верхнє, нижнє та праве — 2 см.

Всі сторінки роботи нумеруються послідовно, починаючи з номера 1. Номер розміщується у правому нижньому куті аркуша. На першій сторінці (титульному аркуші) та листі завдання номер не ставиться. Тому перша нумерована сторінка матиме номер 4.

3.3. Оформлення тексту

Всі текстові елементи записки оформлюються наступним чином: шрифт Times New Roman, розмір 14 пунктів, полуторний міжрядковий інтервал, колір чорний, фон — білий. Інші шрифти дозволяється використовувати лише в ілюстраціях. Для абзаців тексту, нумерованих та маркірованих списків встановлюється вирівнювання тексту по ширині, абзацний відступ — 5 знаків ($\approx 1,25$ см).

3.4. Зміст

Зміст роботи створюється виключно засобами автоматичного генерування змісту текстового редактора і повинен містити посилання на структурні елементи роботи, включаючи номери сторінок.

Вимоги щодо оформлення елементів змісту:

- номери сторінок вирівнюються вправо, проміжки між назвами елементів та номерами заповнюються крапками;
- для елементів 1 рівня: абзацний відступ відсутній, шрифт нормальний, всі літери прописні;
- для елементів 2 рівня: відступ зліва 0,5–1,0 см, шрифт нормальний;
- не можна використовувати напівжирний, нахилений та інші засоби шрифтового оформлення елементів змісту.

3.5. Заголовки

Всі структурні елементи роботи 1-го рівня (зміст, перелік умовних позначень, вступ, розділи, висновки, додатки, список використаних джерел)

повинні починатись з нової сторінки. Структурні елементи нижчих рівнів починаються відразу після завершення попереднього елемента.

Кожен структурний елемент роботи повинен починатись із заголовка — спеціального окремого абзацу, який не може відриватись від наступного за ним абзацу тексту (розташовуватись на іншій сторінці).

Вимоги щодо оформлення заголовків:

- Заголовки 1-го рівня: вирівнювання по центру, текст напівжирний, всі прописні. Абзац починається з нової сторінки, абзацний відступ відсутній. Інтервал після абзацу — не менше 30 пунктів.
- Заголовки 2-го та нижчих рівнів: вирівнювання по ширині, текст напівжирний. Абзацний відступ — ідентичний відступу абзаців тексту роботи. Інтервали перед та після абзацу — по 6 пунктів.
- Крапки в кінці заголовків не ставляться.
- Розділи роботи мають бути пронумеровані таким чином: після слова "РОЗДІЛ" ставиться його номер арабською цифрою, після номера крапка не ставиться, потім з нового рядка друкується заголовок розділу.

Підрозділи роботи 2-го рівня нумеруються таким чином: "Х.У. Заголовок підрозділу", де Х — номер розділу, У — номер підрозділу в межах розділу. Після номеру ставиться крапка.

Заголовки 1-го та 2-го рівнів обов'язково повинні бути відображені у змісті роботи.

3.6. Таблиці

Кожна таблиця повинна мати номер та назву, які розміщуються перед таблицею і оформлюються таким чином: спочатку з вирівнюванням вправо вказується номер у форматі "Таблиця Х.У.", де Х – номер розділу (додатку), У – номер таблиці в межах розділу (додатку); далі з нового рядка вказується назва, яка вирівнюється по центру (крапка в кінці назви не ставиться).

Після назви з нового рядка розміщується сама таблиця, яка не може відриватись від назви та номеру (розміщуватись на іншій сторінці). Таблиця обов'язково повинна мати заголовки колонок, які описуються у першому рядку таблиці ("шапка" таблиці).

Якщо таблиця не може бути розташована на одній сторінці, то її поділяють на частини, кожна з яких розміщена на окремій сторінці та починається з рядка заголовку з назвами стовпців. Кожній частині таблиці (на початку чергової сторінки) передусь запис "Продовж. табл. Х.У."

Вимоги щодо оформлення тексту у таблицях:

- Абзацний відступ відсутній.
- Текст заголовків та підзаголовків – напівжирний.
- Заголовки потрібно починати з великих літер, підзаголовки – з маленьких, якщо вони складають одне речення із заголовком, і з великих, якщо вони незалежні. Крапки в кінці не ставлять.
- Вертикальне та горизонтальне вирівнювання комірок заголовків – по центру.

– Вертикальне вирівнювання усіх інших комірок – вгору, горизонтальне – вліво.

– Для оформлення великих таблиць у додатках дозволяється зменшувати розмір шрифту (але він не повинен бути меншим від 10 пунктів) та робити одинарний міжрядковий інтервал.

Всі комірки таблиці повинні мати границі у вигляді чорної суцільної лінії товщиною 0,5 пунктів.

Ширина таблиці не повинна бути меншою 50 % та більшою 100 % ширини робочого поля аркуша.

Горизонтальне вирівнювання всієї таблиці — по центру. Приклади таблиць наведені у Додатку В.

3.7. Ілюстрації

До ілюстрацій належать рисунки, схеми, графіки, діаграми. Кожна ілюстрація повинна бути підписана як рисунок.

Кожен рисунок повинен мати підпис, що містить номер та назву. Він розміщується в окремому рядку відразу після рисунку і має формат "Рис. Х.У. Назва", де Х – номер розділу (додатку), У – номер рисунку в межах розділу (додатку). Підпис не може відриватись від самого рисунку (розміщуватись на іншій сторінці).

Вимоги щодо оформлення рисунків та підписів: вирівнювання – по центру, без абзацу; шрифт нормальний; крапка в кінці назви рисунку не ставиться.

Рисунки і таблиці варто наводити в роботі безпосередньо після тексту за першою згадкою або на наступній сторінці. Якщо вони розміщені на окремих сторінках роботи, включають до загальної нумерації.

У текст курсової роботи повинні бути вставлені такі рисунки (див. п. 2.5.3):

- Структурна блок-схема алгоритму роботи програми (обов'язково).
- Блок-схема алгоритму (блок-схеми алгоритмів) функцій, що складають основну розробку автора роботи (не обов'язково).
- Рисунки, що відображують результати роботи програми (наприклад екранні форми з заставкою, головним меню, діалогом з користувачем, тощо).

Приклади рисунків наведені у Додатку В.

3.8. Додатки

Додатки повинні мати порядковий номер та назву.

Номер потрібно вирівняти по центру і записати у форматі "Додаток Х", де Х – власне номер (додатки послідовно нумеруються прописними літерами українського алфавіту, починаючи з А, причому І, Є, І, Ї, О, Ч, Ь не використовують). Якщо робота має один додаток, його потрібно позначити як Додаток.

Назва додатку вирівнюється по центру, починається з нового рядка відразу після номера, містить довільний опис змісту додатку. Шрифт – напівжирний.

Кожен додаток потрібно починати з нової сторінки. Посилання на кожен додаток у зміст роботи включати не потрібно.

3.9. Поради щодо оформлення тексту

Пояснювальна записка має бути оформлена відповідно до описаних вище загальних вимог, які є скороченим переказом відповідних стандартів, їх дотримання є обов'язковим.

Описані нижче правила сприятимуть візуально акуратному оформленню тексту. Це:

- не допускати у тексті подвійних пропусків, пропусків перед початком та після абзацу, перед знаками пунктуації (крапками, комами, двокрапками тощо);
- не допускати так званих "висячих" рядків — коли на сторінці розміщується лише один рядок абзацу, а решта — на іншій;
- по всьому тексту роботи використовуються однакові типи маркерів для маркірованих списків, однакові типи лапок (наприклад, «текст», “текст”, „текст” або "текст" – але обов'язково подвійні);
- стиль "курсив" у тексті може використовуватись для позначення окремих фрагментів тексту, на які автор хоче звернути увагу. Крім того, курсив може використовуватись для виділення при першому згадуванні прізвищ осіб, що згадуються в роботі, назв організацій чи підприємств та специфічних термінів;
- стиль "підкреслений" у тексті зазвичай не використовується, зокрема, слід уникати його при позначенні посилань на інтернет-ресурси.

4. ЗАХИСТ РОБОТИ

4.1. Документи, які мають бути підготовлені перед захистом

Перелік матеріалів, які студент повинен подати на кафедру та на пошту ivan734@i.ua, перед захистом курсової роботи:

- електронний варіант пояснювальної записки (у форматі Microsoft Office Word *.doc або *.docx);

- електронний варіант власне програмної розробки курсової роботи (у вигляді окремого каталогу з текстовими файлами модулів з розширенням *.asm, *.obj, *.h, *.cpp, або інших робочих файлів, необхідних для роботи, та виконуваний .exe-файл).

- зброшурований друкований примірник пояснювальної записки до курсової роботи, який повинен містити на титульному аркуші підпис здобувача, резолюцію "До захисту" та підпис наукового керівника;

Відсутність будь-яких з перелічених документів на кафедрі є підставою для недопущення студента до захисту курсової роботи.

4.2. Процедура захисту

Захист курсових робіт проводиться на відкритому засіданні спеціально створеної комісії.

Комісії із захисту курсових робіт подаються усі документи, підготовлені перед захистом (див. п. 4.1) та залікові книжки.

Дозвіл на захист курсової роботи без залікової книжки може бути надано деканом факультету на підставі мотивованої заяви здобувача.

Процедура захисту передбачає:

- доповідь здобувача про зміст роботи, який супроводжується демонстрацією розробленого програмного продукту;

- запитання до автора;

- відповіді здобувача на запитання членів комісії;

- оголошення відгуку наукового керівника;

- заключне слово здобувача;

- оголошення рішення комісії про оцінку роботи.

Доповідь студент повинен підготувати заздалегідь у формі виступу тривалістю до 10 хвилин, в якому повинні бути висвітлені такі питання:

- обґрунтування актуальності теми;

- мета, завдання, стисла характеристика особливостей реалізації;

- демонстрація роботи програми з усними коментарями;

- висновки по роботі;

- відповіді на основні зауваження наукового керівника;

- висновки щодо набутих і вдосконалених під час виконання роботи

4.3. Критерії оцінювання

Оцінка за курсову роботу студента складається з балів, які він отримує за самостійне виконання курсової роботи у відповідності до теми, сформульованій у завданні.

Виконання курсової роботи передбачає:

1) Створення працюючого програмного продукту, який би:

- відповідав призначенню, визначеному обраною темою;
- забезпечував потрібну функціональність та мав розвинений інтерфейс для доступу користувача до означеної функціональності на рівні вимог, перелічених у бланку завдання;

- мав структуру, яка відповідає загальним вимогам до програмного продукту, визначених п.п. 1.3, 1.4 даних Методичних вказівок.

2) Оформлення пояснювальної записки до роботи відповідно стандартам ДСТУ 3008-95;

3) Захист курсової роботи.

Усі етапи виконання роботи повинні виконуватись вчасно у відповідності до затвердженого графіка.

Загальна система балів включає оцінку змісту роботи – програмного продукту (до 50 балів), її оформлення (до 10 балів) та захисту (до 40 балів).

1. Програмний продукт (r1):

- «відмінно», відповідає усім вимогам, визначеним під час постановки задачі, повністю забезпечує потрібну функціональність та дружній до користувача інтерфейс, характеризується стійкістю (захистом від некоректного введення даних та натиснення непередбачуваних клавіш) – 44-50 балів;

- «добре», може бути допущена незначна неточність в реалізації одної з функцій або незначне зауваження до інтерфейсу (наприклад, недостатньо контрастні кольори переднього плану і фону в одному з віконць), або повна відповідь з незначними неточностями – 35-43 бали;

- «задовільно», загалом працююча програма з суттєвими зауваженнями до функціональності, або неврахування вимоги стійкості коду, або недостатньо розвинений інтерфейс (наприклад, такий, що не дає доступу до усіх функціональних можливостей, або використання термінології, не зрозумілої ймовірному користувачу, або інше подібне) – 26-34 бали;

- «незадовільно», програма не забезпечує роботу більше 40% потрібних функцій або не має інтерфейсу користувача, або виконана не самостійно – 0 балів (цей факт може бути з'ясований під час захисту або співбесіди під час демонстрації працездатності програми).

– за кожний тиждень запізнення з виконанням графіку нараховується (-1) бал до даного пункту.

2. Оформлення пояснювальної записки (r2):

- «відмінно», відповідає практично усім (90-100%) вимогам ДСТУ щодо текстового матеріалу, містить потрібний ілюстративний та графічний матеріал, лістинг програмних модулів повний та має змістовні коментарі – 9-10 балів;

– «добре», вимоги ДСТУ виконано на рівні 75-90%, або робота має зауваження щодо повноти забезпечення ілюстративним матеріалом або до якості виконання даного матеріалу – 6-8 балів;

– «задовільно», робота може мати значні зауваження щодо якості текстового матеріалу та повноти забезпечення ілюстративним матеріалом (крім обов'язкових складових) – 1-5 бал;

– «незадовільно», робота не має необхідних складових, або не відповідає більшості вимог стандарту ДСТУ, автору запропоновано переробити пояснювальну записку і далі діяти відповідно загального графіку перездач заборгованостей;

– за кожний тиждень запізнення зі здачею готової пояснювальної записки оцінка знижується на 2 бали.

3. Захист курсової роботи (г3):

– «відмінно», студент демонструє вільне володіння теоретичним матеріалом при показі готового продукту, відповідає на усі додаткові теоретичні і практичні запитання стосовно своєї роботи – 30-40 балів;

– «добре», під час відповідей на теоретичні питання можливі незначні неточності (до 75%) – 20-30 балів;

– «задовільно», можливі неправильні відповіді, які свідчать про виконання роботи з використанням деяких готових алгоритмів без їх творчого опрацювання – 5-20 балів;

– «незадовільно», студент не може продемонструвати готовий програмний продукт, не може його коректно прокоментувати, що свідчить про роботу у співавторстві – 0 балів.

Максимальна сума балів складає 100.

Шкали оцінювання курсової роботи наведені в табл. 2.

Таблиця 2 – Шкали оцінювання результатів курсової роботи

100-бальна шкала	Оцінка курсової роботи	Шкала ECTS
90 – 100	відмінно	A
80 – 89	добре	B
70 – 79		C
66 – 69	задовільно	D
60 – 65		E
0 – 59	незадовільно з можливістю повторного виконання та захисту курсної роботи	FX

Після обговорення результатів публічного захисту комісія оголошує результати підсумкового оцінювання, які заносяться у відомість захисту курсової роботи з підписами членів комісії.

У разі отримання здобувачем вищої освіти за результатами підсумкового оцінювання курсової роботи до 59 балів (включно) у здобувача формується академічна заборгованість. За рішенням комісії здобувач може отримати право:

– повторно захистити курсову роботу в визначену комісією дату до початку екзаменаційної сесії;

– повторно захистити курсову роботу за визначеною комісією темою під час засідання комісії наступного навчального року.

Здобувач, який виконує курсову роботу на умовах повторного виконання та захисту, отримує нове завдання і зобов'язаний дотримуватися сформованого відповідною кафедрою регламенту.

5. РЕКОМЕНДОВАНІ ІНФОРМАЦІЙНІ ДЖЕРЕЛА

5.1. Основна література

1. Ирвин Кип. Язык ассемблера для процессоров INTEL / Ирвин Кип. – 4-е изд.; пер. с англ. – М.: Издат. дом Вильямс, 2005. – 912 с.
2. Кравець В. О. Системне програмування. Програмування на мові асемблера: навч. посіб. / В.О.Кравець, О.М. Рисований. – Х.: НТУ «ХШ», 2007. – 448 с.
3. Кравець В. О. Системне програмування. Асемблер під WIN32 API: навч. посіб. / В. О. Кравець, О. М. Рисований. – Х.: НТУ «ХШ», 2008. – 512 с.
4. Системне програмування: Метод. вказівки до самостійної роботи студентів напряму підготовки 6.050101 «Комп'ютерні науки» / Уклад.: І.В. Назарчук, О.Л. Тимощук – К. ННК «ІПСА» НТУУ «КП», 2012. – 23 с.
5. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення: ДСТУ 3008-95 (ГОСТ 7.32-91). – Чинний від 2006-01-01. К.: Держспоживстандарт України, 2006.– 231с. – (Національний стандарт України).
6. ДСТУ ГОСТ 7.1.2006 «Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання».
7. ДСТУ ГОСТ 7.1:2006 библиографический список, примеры оформления//Известия вузов. Радиоэлектроника. – Режим доступа: <http://radio.kpi.ua/author/rules/11-author-new-biblio>. — Дата доступа: 25.04.10. – Название с экрана.
8. Схеми алгоритмів і програм. Правила виконання: ДСТ 19.002-80 ЕСПД.
9. Види програм і програмних документів: ДСТ 19.101-77 ЕСПД.

5.2 . Додаткова література

1. Крупник А. Ассемблер. Самоучитель / А. Крупник. – С.Пб.: Питер, 2005. – 235 с.
2. Магда Ю. С. Ассемблер для процессоров INTEL / Ю.С. Магда. – С.Пб.: Питер, 2006. – 410 с.
3. Пирогов В. Ю. Ассемблер и дизассемблирование/В.Ю. Пирогов. – С.Пб.: БХВ-Петербург, 2006. – 464 с.
4. Рисований О. М. Системне програмування. Основи асемблера під WIN32 API: навч. посіб. / О.М. Рисований. – Х.: Сучасний Друк, 2008. – 552 с.
5. Рисований О. М. Цифрові пристрої і мікропроцесори. Архітектура і програмне забезпечення: навч. посіб. / О. М. Рисований, М. В. Грушенко. - Х. : ХУПС, 2005. - 384 с.
6. Рисований О. М. Системне програмування [Текст]: підручник для студентів напряму "Комп'ютерна інженерія" вищих навчальних закладів / О. М. Рисований. -Х. : НТУ «ХШ», 2010. - 912 с.

7. Методичні вказівки до виконання та оформлення курсового проекту з курсу «Системне програмування» для студентів спеціальностей: 7.05010201 «Комп'ютерні системи та мережі», 7.05010202 «Системне програмування», 7.05010203 «Спеціалізовані комп'ютерні системи» / уклад. О. М. Рисований - Х. : НТУ «ХШ», 2013. - 184 с.
8. Рогожина С. І. Системне програмування та операційні системи : навч. посіб. Ч. II / С. І. Рогожина, О. М. Рисований, В. М. Федорченко. – Х.: ХУПС, 2007. – 200 с.
9. Щупак Ю. А. Шіп32 API. Эффективная разработка приложений/ Ю.А. Щупак. – С.Пб.: Питер, 2007. – 572 с.

5.3. Дистанційні курси та інформаційні ресурси

1. Системне програмування-Іванченко-INIS15100U [Електронний ресурс]: дистанційний курс Moodle для студентів / Г.Ф. Іванченко. – К.: КНЕУ, 2020. – Режим доступу: <http://193.110.163.11/moodle/enrol/index.php?id=1125>.

ДОДАТКИ

Бланк титульної сторінки курсової роботи

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені ВАДИМА ГЕТЬМАНА»**

Інститут інформаційних технологій в економіці
Кафедра інформаційних систем в економіці

КУРСОВА РОБОТА

з навчальної дисципліни

СИСТЕМНЕ ПРОГРАМУВАННЯ

на тему: надрукувати свою тему

Студента (ки) 1 курсу групи ІКН-1,2
Спеціальність 122 Комп'ютерні науки.
Петренко Петро Петрович

Керівник професор *Іванченко Г.Ф.*

Національна оцінка
Кількість балів:___ Оцінка: ECTS

Члени комісії _____ (підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

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

(підпис) (вчене звання, науковий ступінь, прізвище та ініціал)

Київ – 20XX рік

Зразок оформлення бланку завдання на курсову роботу

ДВНЗ «КНЕУ ім. В. Гетьмана»					
Інститут інформаційних технологій в економіці					
Кафедра	Кафедра інформаційних систем в економіці				
Дисципліна	<i>Системне програмування</i>				
Освітньо-професійна програма	<i>122 «Комп'ютерні науки»</i>				
Курс	<i>перший</i>	Група	<i>ІКН—ХХ</i>	Семестр	<i>другий</i>

ЗАВДАННЯ на курсову роботу студента

(прізвище, ім'я, по батькові)	
1. Тема роботи	
2. Строк здачі студентом закінченої роботи	<i>ХХ.05.20ХХ р.</i>
3. Вихідні дані до роботи	
4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)	
<i>1. Постановка задачі.</i>	
<i>2. Метод розв'язку задачі</i>	
<i>3. Загальна блок-схема алгоритму та опис алгоритму</i>	
<i>4. Опис програмного продукту.</i>	
<i>5. Результати роботи.</i>	
<i>6. Висновки.</i>	
<i>7. Список використаної літератури.</i>	
<i>Додаток А. Текст програми.</i>	
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)	
<i>1. Загальна блок-схема алгоритму.</i>	
<i>2. Ілюстрації роботи програми.</i>	
6. Дата видачі завдання	

КАЛЕНДАРНИЙ ПЛАН

№/п	Назва етапів курсової роботи	Строк виконання етапів роботи	Примітка
1.	Вибір теми курсової роботи. Опрацювання літератури. Оформлення листа Завдання.	2,3-й тиждень лютого	
2.	Аналіз постановки задачі. Узгодження з керівником попереднього плану роботи	3,4-й тиждень лютого	
3.	Вибір та дослідження методів та структур даних, Розробка загального алгоритму	1-й тиждень березня	
4.	Перше узгодження з керівником.	1-й тиждень березня	
5.	Проектування інтерфейсу. Розробка алгоритмів окремих блоків.	2-й тиждень березня	
6.	Друге узгодження з керівником.	2-й тиждень березня	
7.	Програмна реалізація.	3,4-й тижні березня	
8.	Демонстрація першого варіанту. Трете узгодження з керівником.	1-й тиждень квітня	
9.	Доопрацювання програми. Заключне тестування програми.	2,3-й тижні квітня	
10.	Аналіз результатів. Оформлення звіту.	до 2-го тижня травня	
11.	Захист та демонстрація курсової роботи.		

Студент		
	(підпис)	

Керівник			
	(підпис)		(прізвище, ім'я, по батькові)

(дата)	

Зразки таблиць, малюнків, схем

Таблиця Приклад оформлення програми з описом функцій

```
.386 ; директива визначення типу мікропроцесора
.model flat, stdcall ; задання лінійної моделі пам'яті та угоди ОС Windows
option casemap:none ; відмінність малих та великих літер
include \masm32\include\windows.inc ; файли структур, констант ...
include \masm32\include\user32.inc ; файли інтерфейсу ...
include \masm32\include\kernel32.inc ; файли систем. функцій застосувань...
includelib \masm32\lib\user32.lib
includelib \masm32\lib\kernel32.lib
; Для функції Y=40X+15 одержати перше значення, при якому
; Y > 512, ; починаючи з X = 2. Значення аргументу та функції
; записати в комірки пам'яті. Вивести результат на екран
BSIZE equ 15 ; задання реальної кількості байтів
.data ; директива визначення даних
x dd 40 ; резервування 32-розрядної комірки пам'яті для x = 40
y dd ? ; резервування 32-розрядної комірки пам'яті для змінної Y
buf db BSIZE dup(?) ; резервування пам'яті для буфе
frmt db "%d",0 ; задання перетворення одного символу
stdout DWORD ? ; резервування в пам'яті 32-розрядної комірки
cWritten DWORD ? ; резервування 32-розрядної комірки пам'яті
; з ім'ям cWritten для адреси символів виведення
st1 db " y = 40X + 15",0
st2 db "Автор Студент КНЕУ.",0
st3 db " Курсова робота Організація введення-виведення",0
st4 db "Програма працює",0
st6 db "Слава Україні !!!",0
.code ; початок сегмента даних
start: ; мітка початку програми
mov ebx,2 ; початкове значення аргументу X
m1: mov eax,ebx ; підготування до множення
mul x ; виконання частини функції Y (40X)
add eax,15 ; виконання всієї функції Y = 40X + 15
cmp eax,512 ; порівняння з умовою
jnc exit ; перейти, якщо Y > 128
inc ebx ; збільшити лічильник аргументу X
jmp m1 ; перейти при Y < 128
exit:
mov esi,ebx ; збереження X
mov edi,eax ; збереження Y
invoke GetStdHandle, STD_OUTPUT_HANDLE ; отримання дескриптора
```

```

mov stdout, eax ; збереження одержаного дескриптора у пам'яті
invoke wsprintf, \ ; API-функція перетворення числа
ADDR buf, \ ; адреса буфера, куди буде записана послідовність символів
ADDR frmt, \ ; адреса рядка перетворення формату
esi ; регістр, вміст якого перетворюється
invoke WriteConsoleA, stdout, \ ; функція виведення на екран та дескриптор
ADDR buf, \ ; адреса початку повідомлення
BSIZE, \ ; розмір повідомлення
ADDR cWritten, 0 ; адреса, де зберігається число символів
invoke wsprintf, ADDR buf, ADDR frmt, edi ; функція перетворення edi
invoke WriteConsoleA, stdout, \ ; виведення на екран та дескриптор
ADDR buf, \ ; адреса початку повідомлення
BSIZE, \ ; розмір повідомлення
ADDR cWritten, 0 ; адреса, де зберігається число символів
invoke Sleep, 2000d ; API-функція затримки зображення
invoke MessageBox, 0, addr st2, addr st1, MB_OKCANCEL ; вигляд діалогового
вікна
xor eax, 1
jz a1
invoke MessageBox, NULL, addr st4, addr st3, MB_ICONINFORMATION
jmp a2
a1: invoke MessageBox, NULL, addr st6, addr st3, MB_OK
a2:
invoke ExitProcess, 0 ; повернення управління ОС та вивільнення ресурсів
end start ; директива закінчення програми з ім'ям start

```

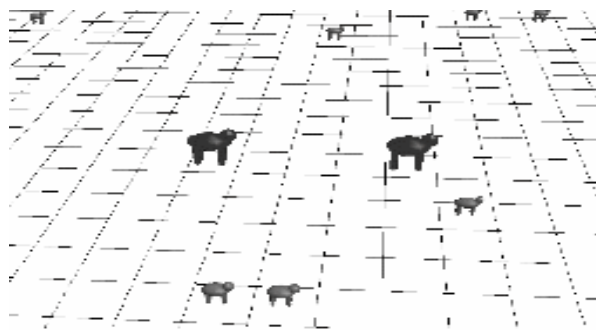


Рисунок 1 – Приклад життєвого простору у моделі «Хижак-Жертва»

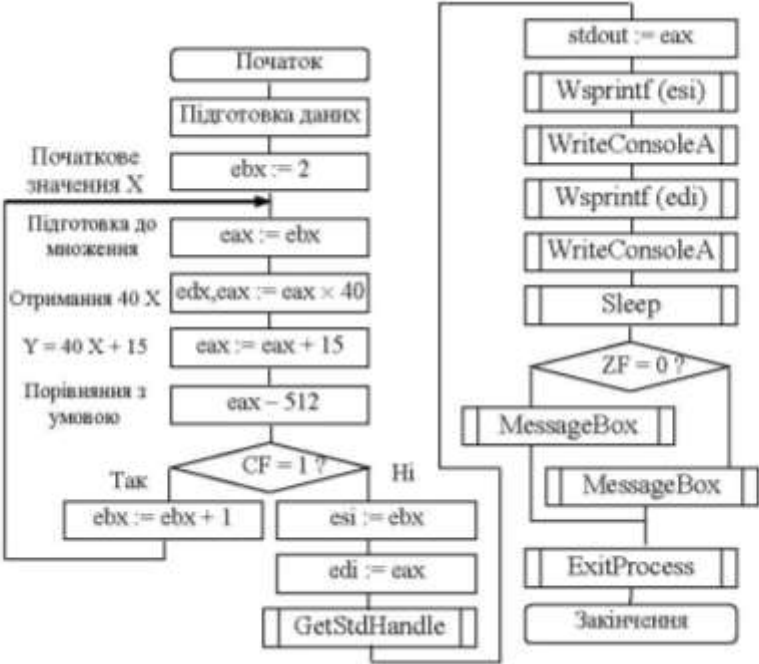


Рисунок 2 – Блок-схема алгоритму

Додаток Г

Приклад оформлення змісту

ВСТУП

РОЗДІЛ 1. Постановка задачі

- 1.1. Огляд існуючих підходів до розв'язання поставленої задачі
- 1.2. Уточнена постановка задачі на розробку програмного забезпечення

РОЗДІЛ 2. Розробка програмного продукту

- 2.1. Метод розв'язку задачі
- 2.2. Алгоритм розв'язку задачі

РОЗДІЛ 3. Опис розробленого програмного продукту

- 3.1. Опис головних структур і змінних програми
- 3.2. Опис головних функцій програми
- 3.3. Налаштування програми з **OLLYDBG**
- 3.4. Опис інтерфейсу
- 3.5. Результати роботи програмного продукту.

ВИСНОВКИ

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

Додаток А Текст програми

Налагоджувач OLLYDBG

Для *виправлення команд в exe-файлі* у налагоджувачі OllyDbg необхідно виконати таке:

- скопіювати програму в нове вікно для корекції. Для цього натиснути праву кнопку миші у вікні дизасемблера і вибрати у нижній частині спливаючого меню пункт Copy to execution / Selection. У результаті весь дизасембльований модуль разом з виправленими командами буде скопійований у нове вікно;
- вказати ім'я нового файлу. Для цього клацнути правою кнопкою миші по новому вікну і вибрати пункт Save file та ввести ім'я нового exe-файлу;
- виконати необхідні зміни в програмі. Для цього підвести курсор до команди, яку необхідно змінити, клацнути на ній, натиснути на праву кнопку миші та обрати пункт Assemble;
- зберегти змінений файл. Для цього натиснути на праву кнопку миші та обрати Copy/To file.

Точки Зупинки

Звичайні точки зупинки

Для того щоб поставити *звичайні точки зупинки* (ordinary breakpoints), необхідно у вікні дизасемблера використовувати клавішу <F2> або подвійне клацання миші в другій колонці вікна коду. У цьому випадку команда забарвлюється в червоний колір. Цей вид зупинки необхідний для того, щоб у точці зупинки можна було б перевірити стан регістрів, змінних, стека. Вторинне натиснення на клавішу <F2> в точці зупинки або подвійне клацання миші видаляють точку зупинки.

Умовні точки зупинки

Умовні точки зупинки (conditional breakpoints) необхідні для того, щоб здійснити зупинку, якщо виконується необхідна умова. Умовні точки зупинки встановлюються натисненням комбінації клавіш <Shift>+<F2>. При цьому з'являється вікно з комбінованим списком, куди можна занести точку зупинки, наприклад, з такою умовою:

- EAX == 5 – зупинка на відміченій команді (перед її виконанням) буде здійснена, якщо вміст регістра EAX дорівнюватиме 5;
- EAX = 0 AND ECX > 6 – зупинка на відміченій команді буде здійснена, якщо вміст регістра EAX дорівнюватиме нулю та вміст регістра ECX буде більше 6;
- [403050] = 1234 – зупинка, якщо вміст елемента пам'яті дорівнює 1234.

Умовні точки зупинки із записом у журнал

Умовні точки зупинки із записом у журнал (conditional logging breakpoint) є розширенням умовних точок зупинки, яке встановлюється натисненням комбінації клавіш <Shift>+<F4>. Для розгляду вмісту журналу необхідно скористатися комбінацією клавіш <Alt>+<L> або вибрати з меню пункт **View | Log**.

Точка зупинки на повідомлення Windows

Для встановлення зупинки на повідомлення Windows необхідно натиснути комбінацію клавіш <Ctrl>+<F8>. Для того щоб вийти на функцію вікна, треба викликати список створених застосувань вікон за допомогою пункту меню **View | Windows**. При роботі з віконними функціями слід ефективніше використовувати точку зупинки на повідомлення. Для цього клацнемо по вікну, яке з'явилося, і виберемо з контекстного меню пункт **Message breakpoint on Classproc**. У вікні, що з'явилося, можна встановити параметри точки зупинки, а саме:

- вибрати повідомлення з випадного списку;
- визначити перелік відстежуваних вікон на предмет надходження даного повідомлення;
- визначити лічильник спрацьовувань точки зупинки;
- встановити чи зупинити виконання програми;
- встановити чи провести запис у журнал.

Точка зупинки на функції імпорту

Список імен, що імпортуються з налагоджуваним модулем, можна отримати за допомогою натиснення комбінації клавіш <Ctrl>+<N>. Далі, клацнувши правою кнопкою миші по вікну, можна встановити:

- точку зупинки на виклик функції, що імпортується (команда **Toggle breakpoint on import**);
- умовну точку зупинки на виклик функції, що імпортується (команда **Conditional breakpoint on import**);
- умовну точку зупинки на імпорт із записом у журнал (команда **Conditional log breakpoint on import**);
- точки зупинки на всі посилання на дане ім'я (команда **Set breakpoint on every reference**);
- точки зупинки із записом у журнал на всі посилання на дане ім'я (команда **Set log breakpoint on every reference**) або видалити всі точки зупинки (команда **Remove all breakpoints**).

Точка зупинки на область пам'яті

Налагоджувач Ollydbg дозволяє встановити одну точку зупинки на область пам'яті. Вибираємо вікно дизасемблера або вікно даних (dump). Далі використовуємо контекстне меню і вибираємо пункт **Breakpoint | Memory on access** (на доступ до пам'яті) або **Breakpoint | Memory on write** (на запис у пам'ять). Після цього точка зупинки готова до використання.

Видалити точку зупинки на область пам'яті можна з контекстного меню **Breakpoint | Remove memory breakpoint**.

Точка зупинки у вікні Memory

Для цієї точки зупинки використовується контекстне меню, що з'являється за допомогою клацання правою кнопкою миші і вибору пункту **Set memory breakpoint on access** (встановити точку зупинки на доступ до пам'яті) або **Set memory breakpoint on write** (встановити точку зупинки на запис у пам'ять). Видалити точку зупинки можна з того контекстного меню командою **Remove memory breakpoint**.

Апаратні точки зупинки

Зазвичай для точки зупинки використовують стандартний вектор переривання int3. У мікропроцесора Intel Pentium є чотири налагоджувальні регістри: DR0 – DR3. Ці регістри можуть містити чотири контрольні точки. Як тільки адреса, яку використовує команда, дорівнює адресі в одному з вказаних регістрів, то тоді генерується виключення, що перехоплюється налагоджувачем. Встановити апаратну точку зупинки можна з вікна дизасемблера за допомогою пункту **Breakpoint | Hardware on execution** контекстного меню або у вікні даних за допомогою пунктів **Breakpoint | Hardware on access** або **Breakpoint | Hardware on access**. Видалити апаратні точки зупинки можна за допомогою контекстного меню **Breakpoint | Remove hardware breakpoints**.

Пошук інформації

Налагоджувач Ollydbg дозволяє ефективно шукати різну інформацію. За командою від натиснення комбінації клавіш <Ctrl>+ з'являється вікно пошуку, де можливо визначити рядок, який розшукуватиметься. Рядок для пошуку можна вводити у вигляді послідовності символів, байтів, символів у кодуванні Unicode.

Для пошуку команд використовуються комбінації клавіш <Ctrl>+<F>

– для одиночної команди і <Ctrl>+<S> – для послідовності команд.

Натиснення комбінації клавіш <Ctrl>+<L> повторює останній зроблений пошук.