

Міністерство освіти і науки України
Вінницький національний технічний університет

**Методичні вказівки
до виконання курсових робіт
з дисципліни «Програмування»
для студентів напряму підготовки
«Автоматизація та комп'ютерно-інтегровані
технології» всіх форм навчання**

Вінниця
ВНТУ
2018

Рекомендовано до видання з грифом «Методичні вказівки ВНТУ (електронне видання)» Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 9 від 30.05.2018 р.)

Рецензенти:

Т. В. Грищук, кандидат технічних наук, доцент

О. М. Васілевський, доктор технічних наук, професор

Ю. В. Булига, кандидат технічних наук, доцент

Методичні вказівки до виконання курсових робіт з дисципліни «Програмування» для студентів напряму підготовки «Автоматизація та комп'ютерно-інтегровані технології» всіх форм навчання / Уклад. Я. А. Кулик, С. М. Довгалець. – Вінниця : ВНТУ, 2018. – 42 с.

У даних методичних вказівках наводяться основні рекомендації для виконання курсових робіт з дисципліни «Програмування» студентами напряму підготовки «Автоматизація та комп'ютерно-інтегровані технології» денної та заочної форм навчання для підготовки бакалаврів з повним на базі ПЗСО та зі скороченим терміном навчання на базі ОКР «Молодший спеціаліст». Наводяться приклади оформлення титульного аркуша, індивідуального завдання, змісту до курсового проекту, що враховують специфіку дисципліни. Методичні вказівки будуть корисними викладачам, аспірантам, магістрантам, студентам, а також всім бажаючим.

Навчальне самостійне електронне мережне видання

Методичні вказівки
до виконання курсових робіт
з дисципліни «Програмування»
для студентів напряму підготовки
«Автоматизація та комп'ютерно-інтегровані технології»
всіх форм навчання

Укладачі: Кулик Ярослав Анатолійович
Довгалець Сергій Михайлович

Електронний ресурс PDF.

Підписано до видання 02.07.2018 р. Зам. № P2018-03

Видавець та виготовлювач - Вінницький національний технічний університет,

Інформаційний редакційно-видавничий центр. ВНТУ, ГНК, к.114,

Хмельницьке шосе, 95, м. Вінниця, 21021,

тел. (0432) 65-18-06.

press.vntu.edu.ua;

Email: irvc.vntu@gmail.com.

Свідоцтво суб'єкта видавничої справи серія ДК № 3516 від 01.07.2009 р.

Вступ

Мета курсової роботи – навчитися застосовувати отриманні теоретичні знання по програмуванню на практиці.

Курсова робота виконується в останньому семестрі вивчення курсу. У результаті виконання курсової роботи повинна бути розроблена програма. Кожна програма повинна мати авторську заставку у графічному режимі.

Текст програми повинен супроводжуватися пояснювальною запискою, яка оформляється відповідно до ГОСТ 19.402-78 «ЕСПД. Р-схеми алгоритмов и программ. Обозначения условные графические и правила выполнения», ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання». У графічній частині обов'язково повинні бути схеми програм, які оформляються відповідно до ГОСТ 19.701-90 (ИСО 5807-85) «ЕСПД. Схеми алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения» та ГОСТ 19.005-85 «ЕСПД. Р-схеми алгоритмов и программ. Обозначения условные графические и правила выполнения».

Зміст курсової роботи

Зміст пояснювальної записки:

1. Титульна сторінка.
2. Технічне завдання.
3. Зміст.
4. Анотація.
5. Вступ.
6. Загальні відомості.
7. Функціональне призначення.
8. Аналіз методів рішення задачі.
9. Опис логічної структури програми.
10. Технічні засоби, що використовуються у розробленій програмі.
11. Виклик та завантаження програми.
12. Вхідні дані.
13. Вихідні дані.
14. Інструкція для користувача.
15. Висновки.
16. Література.
17. Додатки.

Розділи 1, 2, 3, 4, 14, 15 та 16 в тексті не нумеруються. В тексті обов'язково повинні бути посилання на літературу та додатки.

В додатках повинні бути:

- лістинг програми;
- схеми програм;
- тестовий приклад.

Загальні вимоги до курсових проектів

Типовий КР повинен задовольняти такі загальні вимоги:

- обсяг графічної та текстової частин, який визначається кількістю годин СРС, відведеного в навчальному плані дисципліни для КР, за нормами ВНТУ складає три аркуші креслень формату А1, тобто 25...50 сторінок формату А4 текстової частини;

- в графічній частині можуть розроблятися за індивідуальним завданням схеми програми, показуватися зв'язки між модулями, UML-діаграми та діаграми класів, діаграми діяльності та використання.

- при виконанні індивідуальних типових завдань КР за однією принциповою схемою, але різними числовими вхідними даними, текст пояснювальної записки не повинен повторюватися (т.з. електронний трафарет).

До комплексних КР ставляться такі ж вимоги, як і до типових, але загальний обсяг таких проектів збільшується пропорційно кількості студентів, які виконують КР.

Індивідуальне завдання

Конкретний зміст кожної КР, етапи виконання визначає керівник на підставі індивідуального, затвердженого завідувачем відповідної кафедри.

Попередньо керівник видає індивідуальне завдання до курсової роботи. Індивідуальне завдання в перелік змісту не вноситься та має бути другою сторінкою після титульного листа.

Керівник роботи пропонує зміст пояснювальної записки, як правило, в розроблених методичних вказівках або в навчальних цілях зміст може висвітлюватися в індивідуальному завданні.

Кількість варіантів завдань повинна бути такою, щоб вірогідність повторювання варіанта в суміжних групах для дисципліни, яка викладається в межах факультету, була найменшою чи взагалі не існувала (бажано щорічне оновлення варіантів завдань або поєднання різних типів завдань в суміжних навчальних роках), оскільки у студентів можуть залишатися комп'ютерні файли виконаних ними КР.

В залежності від специфіки дисципліни керівник курсової роботи може пропонувати тему, яка підлягає конкретному обґрунтуванню та розробці індивідуального завдання. Індивідуальне завдання до курсової роботи має містити термін видачі, підписи керівника та студента.

Вимоги до оформлення розділів та підрозділів

Структурними елементами основної частини ПЗ є розділи, підрозділи, пункти, підпункти, переліки.

Розділ – головна ступінь поділу тексту, позначена номером і має заголовок.

Підрозділ – частина розділу, позначена номером і має заголовок.

Пункт – частина розділу чи підрозділу, позначена номером і може мати заголовок.

Підпункт – частина пункту, позначена номером і може мати заголовок.

Заголовки структурних елементів необхідно нумерувати тільки арабськими числами.

Допускається розміщувати текст між заголовками розділу і підрозділу, між заголовками підрозділу і пункту.

Кожен розділ рекомендується починати з нової сторінки.

Заголовок розділу записують посередині та з абзацу великими буквами з більш високою насиченістю.

Заголовки розділів, підрозділів, пунктів та підпунктів (при наявності заголовка) записують з абзацу малими буквами починаючи з великої.

Розділи нумерують порядковими номерами в межах всього документа (I, 2, і т.д.). Після номера крапку не ставлять, а пропускають один знак.

Підрозділи нумерують в межах кожного розділу, пункти в межах підрозділу і т.д. за формою (3.1, 3.2, 3.2.1, 3.2.2, 3.2.2.1 і т. д.).

Цифри, які вказують номер, не повинні виступати за абзац.

Посилання в тексті на розділи виконується за формою: "...наведено в розділі 3".

В тексті документа може наводитись перелік, який рекомендується нумерувати малими літерами української абетки з дужкою або тире перед текстом. Для подальшої деталізації переліку використовують арабські цифри з дужкою.

Кожну частину переліку записують з абзацу, починаючи з малої букви і закінчуючи крапкою з комою, в кінці останньої ставлять крапку.

Приклад:

а) текст переліку та його...

продовження;

б) текст переліку:

1) текст переліку подальшої деталізації та його продовження;

2) ...;

в) останній перелік.

Правила написання тексту

При написанні тексту слід дотримуватися таких правил:

а) текст необхідно викладати обґрунтовано в лаконічному технічно-му стилі;

б) умовні буквені позначення фізичних величин і умовні графічні позначення компонентів повинні відповідати установленим стандартам. Перед буквеним позначенням фізичної величини повинно бути її пояснення (*резистор R, конденсатор C*);

в) числа з розмірністю слід записувати цифрами, а без розмірності словами (*відстань – 2 мм, відміряти три рази*);

г) позначення одиниць слід писати в рядок з числовим значенням без перенесення в наступний рядок. Між останньою цифрою числа і позначенням одиниці слід робити пропуск (*100 Вт, 2 А*);

д) якщо наводиться ряд числових значень однієї і тієї ж фізичної величини, то одиницю фізичної величини вказують тільки після останнього числового значення (*1,5; 1,75; 2 мм*);

е) позначення величин з граничними відхиленнями слід записувати так: *100 ± 5 мм*;

ж) буквені позначення одиниць, які входять в добуток, розділяють крапкою на середній лінії (·); знак ділення замінюють косою рисою (/);

и) порядкові числівники слід записувати цифрами з відмінковими закінченнями (*9-й день, 4-а лінія*); при кількох порядкових числівниках відмінкове закінчення записують після останнього (*3,4,5-й графіки*); кількісні числівники записують без відмінкових закінчень (*на 20 аркушах*); не пишуть закінчення в датах (*21 жовтня*) та при римських числах (*XXI століття*);

к) скорочення слів в тексті не допускаються, крім загальноприйнятих в українській мові і установлених в ДСТУ 3582:2013, а також скорочень, які прийняті для надписів на виробі (в тексті вони повинні бути виділені великими літерами: ON, OFF), а якщо надпис складається з цифр або знаків, то в лапках. Лапками також виділяють найменування команд, режимів, сигналів (“*Запуск*”);

л) дозволяється виконувати записи математичних виразів за формою:

$$\frac{ABC}{DE} = ABC/DE ;$$

знак множення “×” замінювати зірочкою “*” (ГОСТ 2.004-88).

м) не дозволяється:

- допускати професійних або місцевих слів і виразів (техніцизмів);
- після назви місяця писати слово “*місяць*” (не “*в травні місяці*”, а “*в травні*”);

- використовувати вирази: “*цього року*”, “*минулого року*”, слід писати конкретну дату “*в червні 2001 року*”;

- використовувати позначення одиниць фізичних величин без цифр, необхідно писати повністю: “*кілька кілограмів*” (за винятком оформлення таблиць і формул);

- з'єднувати текст з умовним позначенням фізичних величин за допомогою математичних знаків (не “*швидкість = 5 км/год*”, а “*швидкість дорівнює 5 км/год*”, не “*температура дорівнює - 5° С*”, а “*температура дорівнює мінус 5° С*”);

- використовувати математичні знаки <, >, o, №, %, sin, cos, tg, log та ін. без цифрових або буквених позначень. В тексті слід писати словами “*нуль*”, “*номер*”, “*логарифм*” і т.д.;

- використовувати індекси стандартів (ДСТУ) без реєстраційного номера.

Оформлення формул

Кожну формулу записують з нового рядка, симетрично до тексту. Між формулою і текстом пропускають один рядок.

Умовні буквені позначення (символи) в формулі наводять в тексті або зразу ж під формулою. Для цього після формули ставлять кому і записують пояснення до кожного символу з нового рядка в тій послідовності, в якій вони наведені у формулі, розділяючи крапкою з комою. Перший рядок повинен починатися з абзацу з слова “де” і без будь-якого знака після нього.

Всі формули нумерують в межах розділу арабськими числами. Номер вказують в круглих дужках з правої сторони, в кінці рядка, на рівні закінчення формули. Номер формули складається з номера розділу і порядкового номера

формули в розділі, розділених крапкою. Дозволяється виконувати нумерацію в межах всього документа.

Приклад

Таким чином, момент тертя в кернових опорах

$$M_m = - \kappa G^{1,5}, \quad (5.1)$$

де κ – коефіцієнт пропорційності;

G – вага рухомої частини вимірювального механізму.

Одиницю вимірювання, при необхідності, беруть в квадратні дужки

$$I = \frac{U}{R} [A]. \quad (5.2)$$

Числово підстановку і розрахунок виконують з нового рядка не нумеруючи. Одиницю вимірювання беруть в круглі дужки. Наприклад,

$$I = \frac{220}{100} = 2,2 (A).$$

Розмірність одного й того ж параметра в межах документа повинна бути однаковою.

Якщо формула велика, то її можна переносити в наступні рядки. Перенесення виконують тільки математичними знаками, повторюючи знак на початку наступного рядка. При цьому знак множення “ \cdot ” замінюють знаком “ \times ”.

Формула є частиною речення, тому до неї застосовують такі ж правила граматики, як і до інших членів речення. Якщо формула знаходиться в кінці речення, то після неї ставлять крапку. Формули, які йдуть одна за одною і не розділені текстом, відокремлюють комою.

Посилання на формули в тексті дають в круглих дужках за формою: “... в формулі (5.2)”;

Оформлення ілюстрацій

Для пояснення викладеного тексту рекомендується його ілюструвати графіками, кресленнями, фрагментами схем та ін., які можна виконувати чорною тушшю, простим олівцем середньої твердості та комп’ютерною графікою.

Розміщують ілюстрації в тексті або в додатках.

В тексті ілюстрацію розміщують симетрично до тексту після першого посилання на неї або на наступній сторінці, якщо на даній вона не уміщується без повороту.

На всі ілюстрації в тексті ПЗ мають бути посилання. Посилання виконують за формою: “...показано на рисунку 3.1.” або в дужках за текстом (*рисунок 3.1*), на частину ілюстрації: “... показані на рисунку 3.2,б”. Посилання на раніше наведені ілюстрації дають зі скороченим словом “*дивись*” відповідно в дужках (*див.*

рисунок 1.3).

Наведена форма запису (рисунок ...) допускає скорочення, тобто замість „Рисунок ...” – „Рис ...”.

Між ілюстрацією і текстом пропускають один рядок (3 інтервали).

Всі ілюстрації в ПЗ називають рисунками і позначають під ілюстрацією симетрично до неї за такою формою: “Рисунок 3.5 – Найменування рисунка”. Крапку в кінці не ставлять, знак переносу не використовують. Якщо найменування рисунка довге, то його продовжують у наступному рядку починаючи від найменування.

Нумерують ілюстрації в межах розділів, вказуючи номер розділу і порядковий номер ілюстрації в розділі розділяючи крапкою. Дозволяється нумерувати в межах всього документа.

Пояснюючі дані розміщують під ілюстрацією над її позначенням.

У випадку, коли ілюстрація складається з частин, їх позначають малими буквами українського алфавіту з дужкою (а), б)) під відповідною частиною. В такому випадку після найменування ілюстрації ставлять двокрапку і дають найменування кожної частини за формою:

а) – найменування першої частини; б) – найменування другої частини
або за ходом найменування ілюстрації, беручи букви в дужки:

Рисунок 3.2 - Структурна схема (а) і часові діаграми (б) роботи фазометра

Якщо частини ілюстрації не вміщуються на одній сторінці, то їх переносять на наступні сторінки. В цьому випадку, під початком ілюстрації вказують повне її позначення, а під її продовженнями позначають “Рисунок 3.2” (продовження). Пояснюючі дані розміщують під кожною частиною ілюстрації.

Якщо в тексті є посилання на складові частини зображеного засобу, то на відповідній ілюстрації вказують їх порядкові номери в межах ілюстрації.

Якщо ілюстрація є фрагментом повної розробленої схеми, то для всіх компонентів вказують ті позиційні позначення, які вказані на схемі.

Якщо ілюстраціями є фотографії, то останні повинні бути наклеєні на стандартні аркуші білого паперу і позначені як рисунки.

Оформлення таблиць

Таблицю розміщують симетрично до тексту після першого посилання на даній сторінці або на наступній, якщо на даній вона не уміщується і таким чином, щоб зручно було її розглядати без повороту або з поворотом на кут 90° за годинниковою стрілкою.

Запис таблиці:

Таблиця $\frac{\text{---}}{\text{(номер)}} - \frac{\text{---}}{\text{(назва таблиці)}}$

На всі таблиці мають бути посилання за формою: “наведено в таблиці 3.1”; “... в таблицях 3.1 – 3.5” або в дужках по тексту (таблиця 3.6). Посилання на раніше наведену таблицю дають з скороченим словом ”дивись” (див. таблицю 2.4) за ходом чи в кінці речення.

Таблицю розділяють на графи (колонки) і рядки. В верхній частині

розміщують головку таблиці, в якій вказують найменування граф. Діагональне ділення головки таблиці не допускається. Ліву графу (боковик) часто використовують для найменування рядків. Допускається не розділяти рядки горизонтальними лініями. Мінімальний розмір між основами рядків – 8 мм. Розміри таблиці визначаються об'ємом матеріалу.

Графу “№ п/п” в таблицю не включають. При необхідності нумерації, номери вказують в боковикі таблиці перед найменуванням рядка.

Найменування граф може складатися з заголовків і підзаголовків, які записують в однині, симетрично до тексту графи малими буквами, починаючи з великої. Якщо підзаголовок складає одне речення з заголовком, то в цьому випадку його починають з малої букви. В кінці заголовків і підзаголовків граф таблиці крапку не ставлять. Дозволяється заголовки і підзаголовки граф таблиці виконувати через один інтервал.

Якщо всі параметри величин, які наведені в таблиці, мають одну й ту саму одиницю фізичної величини, то над таблицею розміщують її скорочене позначення (мм). Якщо ж параметри мають різні одиниці фізичних величин, то позначення одиниць записують в заголовках граф після коми (*Довжина, мм*).

Текст заголовків і підзаголовків граф може бути замінений буквеними позначеннями, якщо тільки вони пояснені в попередньому тексті чи на ілюстраціях (*D – діаметр, H – висота і т.д.*). Однакові буквені позначення групують послідовно в порядку росту їх індексів, наприклад: (*L₁, L₂, ...*).

Найменування рядків записують в боковикі таблиці у вигляді заголовків в називному відмінку однини, малими буквами, починаючи з великої і з однієї позиції. В кінці заголовків крапку не ставлять. Позначення одиниць фізичних величин вказують в заголовках після коми.

Для опису визначеного інтервалу значень в найменуваннях граф і рядків таблиці можна використовувати слова: “більше”, “менше”, “не більше”, “не менше”, “в межах”. Ці слова розміщують після одиниці фізичної величини:

(*Напруга, B, не більше*),

а також використовують слова “від”, “більше”, “до”:

(*Від 10 до 15; більше 15; до 20*)

Дані, що наводяться в таблиці, можуть бути словесними і числовими.

Слова записують в графах з однієї позиції. Якщо рядки таблиці не розділені лініями, то текст, який повторюється і складається з одного слова дозволяється замінювати лапками (,). Якщо текст складається з двох і більше слів, то при першому повторенні його замінюють словами “те ж”, а далі лапками. При розділенні таблиці горизонтальними лініями – ніякої заміни не виконують.

Числа записують посередині графи так, щоб їх однакові розряди по всій графі були точно один під одним, за виключенням випадку, коли вказують інтервал. Інтервал вказують від меншого числа до більшого з тире між ними:

12 – 35
122 – 450.

Дробові числа наводять у вигляді десяткових дробів, з однаковою кількістю знаків після коми в одній графі. Розміри в дюймах можна записувати у вигляді: $1/2''$, $1/4''$, $1/8''$.

Ставити лапки замість цифр чи математичних символів, які повторюються не можна. Якщо цифрові чи інші дані в таблиці не наводяться, то ставиться прочерк.

Таблиці нумерують в межах розділів і позначають зліва над таблицею за формою: “Таблиця 4.2 – Найменування таблиці”. Крапку в кінці не ставлять. Якщо найменування таблиці довге, то продовжують у наступному рядку починаючи від слова “Таблиця”. Номер таблиці складається з номера розділу і порядкового номера таблиці в розділі, розділених крапкою. Дозволяється нумерувати в межах всього документа.

Таблиця може бути великою як в горизонтальному, так і в вертикальному напрямках або іншими словами може мати велику кількість граф і рядків. В таких випадках таблицю розділяють на частини і переносять на інші сторінки або розміщують одну частину під іншою чи поряд.

Якщо частини таблиці розміщують поряд, то в кожній частині повторюють головку таблиці, а при розміщенні однієї частини під іншою – повторюють боковик.

Якщо в кінці сторінки таблиця переривається і її продовження буде на наступній сторінці, в першій частині таблиці нижню горизонтальну лінію, що обмежує таблицю, не проводять.

У випадку перенесення частин таблиці на інші сторінки, повторюють або продовжують найменування граф. Допускається виконувати нумерацію граф і на початку таблиці та, у випадку перенесення частин таблиці на наступні сторінки повторювати тільки нумерацію граф.

У всіх випадках найменування (при його наявності) таблиці розміщують тільки над першою частиною, а над іншими частинами зліва пишуть “Продовження таблиці 4.2” без крапки в кінці.

Інші вимоги до виконання таблиць – відповідно до чинних стандартів на технічну документацію.

Зміст

Зміст розташовують безпосередньо після анотації, починаючи з нової сторінки. До змісту включають: перелік умовних позначень, символів, одиниць, скорочень і термінів; вступ; послідовно перелічені назви всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають заголовки) суті роботи; висновки; рекомендації; перелік посилань; назви додатків і номери сторінок, які містять початок матеріалу. У змісті можуть бути перелічені номери й назви ілюстрацій та таблиць з зазначенням сторінок, на яких вони вміщені.

Зміст за нумерацією пояснювальної записки є третьою сторінкою, на якій для курсового проекту виконують основний надпис за формою 2 (40×185 мм), на наступних – за формою 2а (15×185 мм).

Назви заголовків змісту повинні однозначно відповідати назвам заголовків пояснювальної записки за текстом. Нумерація сторінок повинна бути наскрізною.

Форми подачі розділів та підрозділів в змісті для курсових робіт (КР) показані нижче.

КР

1 Аналіз ...

1.1 Огляд ...

1.1.1 ...

2 Заголовок другого розділу

2.1 Заголовки підрозділів

2.1.1 ...

3 Заголовок третього розділу

3.1 Заголовки підрозділів

3.1.1 ...

При виконанні курсових робіт обсяг пояснювальної записки враховується до додатків. Якщо додатки курсових робіт підтверджують цінність результату проектування, то обсяг пояснювальної записки з додатками повинен мати наскрізну нумерацію.

Складові частини пояснювальної записки

Пояснювальна записка повинна відповідати індивідуальному завданню, а її оформлення – чинним стандартам, які слід враховувати на момент виконання розробки з врахуванням всіх офіційних змін, введених в дію.

Конкретний зміст пояснювальної записки до КР (вихідні дані та перелік питань, які підлягають розробці) визначає керівник, обов'язковими складовими якої є:

- вступ;
- аналіз сучасного стану питання та обґрунтування теми, знайомство з положеннями норм;
- основна частина, яка складається із аналітично-розрахункової частини для КР;
- висновки;
- перелік посилань (КР);
- додатки (при необхідності).

“ВСТУП”, “ВИСНОВКИ”, ”ПЕРЕЛІК ПОСИЛАНЬ”, “Додатки” як розділи, не нумеруються.

Текст пояснювальної записки бажано подавати лаконічно та обґрунтовано.

Критерії оцінювання курсової роботи

Для виконання курсової роботи студент отримує завдання від викладача. Також студент може вибрати завдання із даних методичних вказівок або запропонувати власне завдання. Завдання отримує «юридичну силу» після оформлення у вигляді технічного завдання.

Технічне завдання складається студентом та узгоджується із викладачем. У цьому «документі» (розділі КР) прописується вимоги до розроблюваного програмного забезпечення-що воно повинно обов'язково робити, а також бажані(або небажані) технології розробки. Після затвердження тексту технічного завдання студент та викладач ставлять свої підписи. Інтерпретація завдання здійснюється на основі технічного завдання, тому у цього розділі завдання має бути прописане якомога детальніше.

Функціональність та технології розробки програмного забезпечення, які не прописані у технічному завданні, повинні узгоджуватись із викладачем. У випадку виникнення суперечки між викладачем та студентом з приводу оцінювання курсової роботи або інтерпретації функціональності програмного забезпечення та технічного завдання, може бути створення комісія, до якої увійдуть представники від студентства та інші викладачі (або професійні розробники), які є спеціалістами із використаних у КР технологій програмування.

При написанні курсової роботи викладач грає роль замовника, студент-виконавця, технічне завдання-договір про розробку програмного продукту, пояснювальна записка-технічної документації, оцінка за курсову-оплату за виконання замовлення.

При написанні програмного забезпечення дозволяється використання програмних бібліотек від офіційних розробників, програмних бібліотек від сторонніх розробників (репозиторії); використання готових програмних модулів від сторонніх розробників (не програмних бібліотек, а модулів!) та програмного забезпечення повинно узгоджуватись із викладачем та за наступних умов:

- 1) сторонній модуль або програма допомагає реалізації завдання, а не виконує саме завдання(або його більшу частину);
- 2) сторонній модуль або програма повинні мати ліцензію, яка дозволяє копіювання та модифікацію його модулів для своїх задач (ліцензія типу «вільне програмне забезпечення»);
- 3) сторонній модуль або програма повинні бути у відкритому доступі;
- 4) у пояснювальній записці повинно бути чітко вказана, яка частина використані для власної задачі, для яких цілей (реалізації якої дії чи функціональності) і хто автор даних модулів або програм(тобто збереженням авторських прав через посилання у літературі);
- 5) автор стороннього модуля або програми не повинні отримувати грошової чи іншої винагороди за його використання студентом для даної курсової роботи.

Оцінка за курсову роботу може бути підвищена у випадку:

- 1) Здача курсової роботи раніше очікуваного терміну(за місяць до початку сесії).
- 2) Створення функціональності програми, яка не прописана у технічному завданні.
- 3) Використання оригінальних та творчих ідей при реалізації завдання.

- 4) Використання складних технологій для реалізації програми.
- 5) Використання найбільш сучасних технологій для реалізації програми.
- 6) Виконання завдання, яке має реального замовника.
- 7) Виконання завдання, яке може мати потенційних замовників (створення власного «стартапа»).
- 8) Участь у олімпіадах із програмування.
- 9) Розгорнуте написання пояснювальної записки із збереженням посилання на літературу.

Оцінка за курсову роботу може бути знижена у випадку:

- 1) Здача курсової роботи пізніше очікуваного терміну (початку сесії), відставання від термінів, прописаних в таблиці 1.
- 2) Здача курсової роботи пізніше іспиту із предмету «програмування».
- 3) Студент при захисті курсової роботи не може пояснити призначення, дію, спосіб реалізації окремих частин програми, яка реалізує вказане завдання.
- 4) Відсутність функціональності програми, яка прописані у технічному завданні.
- 5) Наявність програмних помилок у виконаному завданні, некоректна робота при певних діях або певних вхідних даних.
- 6) Аргументованих доказів, що студент зробив курсову роботу не самостійно.
- 7) Аргументованих доказів, що студент повністю скопіював фрагмент курсової роботи у іншого студента (в т.ч. з робіт минулих років).
- 8) Погано написана пояснювальна записка: не відповідає вимогам до оформлення; не пояснює структуру програми, призначення всіх модулів, їх функцій; відсутність певних розділів; відсутність огляду використовуваних технологій та подібних існуючих програм.

Таблиця 1 – Графік виконання курсової роботи

Номер етапу	Номер тижня	Назва робіт у етапі	У % від загального
1	1-2	Отримання і уточнення завдання	5
2	3-4	Написання технічного завдання	5
3	4-7	Вивчення теоретичних відомостей предметної області, аналіз літератури	20
4	8	Проектування програм: 1. Обрання структури даних. 2. Обрання форм представлення вхідних і вихідних даних. 3. Вибір мови програмування. 4. Вибір технологій програмування та інструментів реалізації. 5. Розробка алгоритму(схеми програми)	30
5	9-14	Розробка програми	60
6	15	Відлагодження програми, виправлення помилок	75
7	16-17	Оформлення пояснювальної записки.	85
8	18	Захист роботи.	100

Завдання на курсову роботу

Примітки:

С – Завдання від *спеціаліста* в певній галузі. Для виконання даної роботи може знадобитись консультація спеціаліста у певній галузі (забезпечує викладач).

М – Завдання з *математичною* складовою. Для виконання даного завдання потрібні знання у певній галузі математики (у рамках університетського курсу «вищої математики»).

К – *Командне* завдання. Виконання даного завдання передбачає розподіл завдання між кількома виконавцями. Усім виконавцям потрібно домовитися про використання сумісних між собою технологій, способів реалізацій та узгодити спосіб обміну даними між програмними модулями. Оцінювання кожного виконавця буде здійснюватись залежно від якості виконаного власного завдання та незалежно від якості виконаних інших частин завдання, проте за умови поганого виконання завдань іншими учасниками усе завдання разом не функціонуватиме належним чином.

Т – Завдання із використанням *конкретних технологій*. При виконання даного завдання використання даних технологій є обов'язковим. Студент у технічному завданні може запропонувати альтернативну реалізацію даної технології, або запропонувати використання подібних та інших технологій.

П – Завдання для реалізації на *портативних пристроях* (планшети, мобільні телефони). Вибір мобільної системи/платформи/пристрою описується у пояснювальній записці (якщо завданням не передбачено використання чітко вказаних технологій мобільної системи/платформи/пристрою)

Г - Завдання із *графікою*. Завдання потребує приділення великої уваги використанню технологій роботи з графікою або графічному інтерфейсу користувача(GUI).

З – Завдання із *захистом інформації*. Завдання передбачає обов'язковий захист даних користувача від стороннього доступу шляхом використання криптографії, стенографії, аутентифікації користувача, цифрового підпису та інших технологій захисту інформації.

В – Завдання передбачає створення *веб-сайту*. Призначення, основні функції та контент веб-сайту повинні описуватися у технічному завданні.

Б – Завдання із *базою даних*. Завдання передбачає обов'язкове використання бази даних. Вибір бази даних та супутніх технологій(сервер, протокол та мова запитів, інше) повинен описуватися у пояснювальній записці.

Під складністю виконання завдання мається на увазі очікуваний рівень знань, які потрібні студенту для написання курсової роботи.

Теми курсових робіт

Задачі, які часто рішення яких часто потребують науковці та спеціалісти певних галузей

Захист інформації

1. Реалізація шифрування інформації вказаним шифром (3 способи).

2. Реалізація шифрування інформації вказаним шифром із використанням фреймворку Spring.
3. Реалізація генератора випадкових чисел та аналізатора випадковості.
4. Реалізація програми, яка перевіряє цифровий підпис.
5. Реалізація шифру Віженера та простої підстановки.
6. Реалізація геометричного шифру.

Різні задачі

7. Реалізація програми для балістичної експертизи.
8. Реалізація програми для відображення дерева версій Unix з можливістю додавання нових версій.
9. Реалізація запису структурованого тексту в базу даних з можливістю підтримки NoSQL.
10. Реалізація для Linux Ubuntu запису форматowanego тексту в базу даних.
11. Реалізація будильника з використанням JBoss Seam.
12. Реалізація програми для розрахунку та відображення поточної фази місяця
13. Реалізація мобільної програми для виведення інформації про речовину за введеним E-кодом.
14. Реалізація програми, яка переводить цифровий код у штрих-код і визначає виробника товару.
15. Реалізація програми, яка переводить цифровий код у QR-код і визначає виробника товару.
16. Реалізація програми для малювання української вишивки, характерної для Вінниччини.
17. Реалізація програми для малювання української вишивки за способом «Бродівське письмо».

Мобільні пристрої

18. Реалізація будильника для мобільного пристрою.
19. Реалізація програми для упорядкування музики для мобільного пристрою.
20. Реалізація простого диспетчера задач для мобільного пристрою.
21. Реалізація програми для визначення поточного місцезнаходження для мобільного пристрою.
22. Реалізація програми для визначення довжини маршруту від поточного місцезнаходження для мобільного пристрою.

Соцмережі

23. Реалізація автоматичного відповідача у нічний час для соцмережі Facebook.
24. Реалізація програми для автоматичного написання відтермінованих повідомлень у заданий час для соцмережі Twitter.
25. Реалізація програми для автоматичного виставлення фотографій у заданий час для соцмережі Instagram.

Інтернет, локальна мережа

26. Реалізація автоматизованого перекладача з використанням Hibernate.
27. Реалізація програми для відстежування курсу валют із графічним відображенням коливань на основі Spring Web Flow (SWF).
28. Реалізація програми для автоматичної відправки повідомлень по електронній пошті у вказаний час за певних умов.
29. Реалізація власного простого завантажувача сайту для автономного перегляду при відсутності доступу до мережі інтернет.
30. Реалізація програми для збереження картинок з веб-сайту на комп'ютер.
Реалізація програми для обміну повідомленнями (месенджера):
31. Частина 1. Реалізація інтерфейсу користувача(GUI).
32. Частина 2. Реалізація обміну текстовими повідомленнями через інтернет та локальну мережу.
33. Частина 3. Реалізація обміну файлами через інтернет та локальну мережу.
34. Частина 4. Реалізація збереження історії повідомлень у базі даних із можливістю пошуку.
Реалізація власної простої програми для доставки файлів за технологією торент-трекер:
35. Частина 1. Реалізація інтерфейсу користувача(GUI).
36. Частина 2. Реалізація простого torrent-клієнта.
37. Частина 3. Реалізація простого torrent-сервера(для роздачі).
38. Частина 4. Реалізація програми для оптимізації швидкості при нестабільній та різній швидкості з'єднання у користувачів.
39. Частина 5. Реалізація програми для оптимізації часу доставки та маршруту проходження пакетів.
40. Частина 6. Реалізація програми для захищеного передавання (криптографічне шифрування пакетів та аутентифікація користувачів).

Веб-технології

41. Створення інформаційного веб-сайту на HTML5.
42. Створення інформаційного веб-сайту з можливістю перегляду фотографій на DHTML.
43. Створення веб-сайту-форуму із БД DB/2 на HTML не нижче 4.0.
44. Створення інформаційного веб-сайту, який здатен працювати з БД.
45. Створення інформаційного веб-сайту із можливістю перегляду відео на JavaFX.
46. Створення веб-сайту для музиканта з можливістю керувати доступом відтворенням композицій залежно від рейтингу користувача на сайті.
47. Створення веб-сайту для перегляду інформації про певний товар зі списку із використанням JavaServer Faces (JSF).
48. Створення веб-сайту, який доступний на відкритому сервері та зберігає інформацію, захищену від хакерських атак (будь-хто зі групи має можливість спробувати викрасти цю інформацію).
49. Створення веб-сайту для масової гри у «хрестики-нулики» онлайн на скриптах(PHP/JavaScript/JScript/ін.).
50. Створення веб-сайту для масової гри у «шашки» онлайн із БД на Google Web Toolkit.

Створення веб-сайту для масової гри у «морський бій» онлайн, який буде надавати можливість грати багатьом користувачам окремо та між собою.

51. Частина 1. Створення веб-сайту із авторизацією користувачів.

52. Частина 2. Створення графічного інтерфейсу гри у «морський бій».

53. Частина 3. Створення штучного інтелекту для гри у «морський бій».

54. Частина 4. Створення інтерфейсу та функцій для ведення статистики на сайті та записом у БД.

Ігри

Реалізація гри «морський бій» із автоматичним встановленням програми(інсталятор)

55. Частина 1. Реалізація інтерфейсу із авторизацією користувачів та можливістю обміну повідомленнями по локальній мережі.

56. Частина 2. Реалізація графічного інтерфейсу гри у «морський бій».

57. Частина 3. Реалізація штучного інтелекту для гри у «морський бій».

58. Частина 4. Реалізація ведення статистики та запису у БД Cloudscape.

59. Частина 5. Реалізація програми встановлення гри із визначенням платформи та перевірки можливості встановлення гри.

60. Реалізація гри «пятнашки» для мобільного пристрою.

61. Реалізація гри «хрестики-нулики» для мобільного пристрою на Android.

62. Реалізація гри «монополія».

Реалізація програми для гри у пінг-понг по локальній мережі.

63. Частина 1. Реалізація графічного інтерфейсу та штучного інтелекту способом 1.

64. Частина 2. Реалізація з'єднання по Internet та штучного інтелекту способом 2.

Реалізація гри «лучник».

65. Частина 1. Реалізація інтерфейсу користувача(GUI).

66. Частина 2. Реалізація модуля для прораховування траєкторії стріли залежно від кута, сили запуску, напрямку та сили вітру.

67. Частина 3. Реалізація модуля для створення перешкод (непробивні блоки, відбивачі, важелі, механізми).

68. Частина 4. Реалізація набору рівнів гри та етапу навчання.

Реалізація гри «мушкетер».

69. Частина 1. Реалізація інтерфейсу користувача(GUI).

70. Частина 2. Реалізація модуля для прораховування траєкторії кулі залежно від кута, кількості пороху, напрямку та сили вітру.

71. Частина 3. Реалізація модуля властивостей матеріалів поверхонь(міцність, крихкість, можливість рикошету та роздроблення) та розрахунок імпульсу кулі до і після влучання.

72. Частина 4. Реалізація модуля штучного інтелекту для ворогів (різних типів).

73. Частина 5. Реалізація модуля вдосконалень для зброї гравця та ворогів.

74. Реалізація програми з віртуальною реальність(VR).

75. Реалізація програми з доповненою реальність(AR).

76. Реалізація програми з доповненою реальність(AR) з допомогою Unity 3D.

77. Реалізація шпигунської програми, замасковану під гру з доповненою реальність(AR).

78. Реалізація гри «симулятор каменю».

79. Реалізація гри «лабіринт».

80. Реалізація гри на основі ігрового рушія Minecraft.

Детальний опис завдань

1. Реалізація шифрування інформації вказаним шифром (3 способи).

Програма повинна реалізувати 3 різні способи шифрування тексту довільної довжини (наприклад 3-DES, AES, ДСТУ ГОСТ 28147:2009), генерувати ключі(відкриті або закриті), які не будуть відомі нікому окрім тих, хто читає повідомлення(для відкритого ключа це не строга умова). Шифр повинен мати крім достатньої стійкості також додаткові способи захисту(наприклад-додавання «криптографічної солі(salt)»). Реалізовувати самі шифри не потрібно.

Вхідні дані: незашифрований текст.

Вихідні дані: зашифрований текст.

Складність: середня.

Примітки: 3.

2. Реалізація шифрування інформації вказаним шифром із використанням фреймворку Spring.

Програма повинна реалізувати 3 різні способи шифрування тексту довільної довжини, генерувати ключі(відкриті або закриті), які не будуть відомі нікому окрім тих, хто читає повідомлення(для відкритого ключа це не строга умова). Шифр повинен мати крім достатньої стійкості також додаткові способи захисту(наприклад-додавання «криптографічної солі(salt)»). Пропонуються на вибір для використання шифри 3-DES, AES, ДСТУ ГОСТ 28147:2009, IDEA, Twofish, BlowFish, TreeFish, SAFER, SHARK, FEAL, KASUMI, RC5, TEA, CRYPTON, Hierocrypt-3, Serpent, Camellia, CAST-256, MARS, RC6. Реалізовувати самі шифри не потрібно. Обов'язкове використання для реалізації програми каркасу Spring.

Вхідні дані: незашифрований текст.

Вихідні дані: зашифрований текст.

Складність: середня.

Примітки: 3, Т.

3. Реалізація генератора випадкових чисел та аналізатора випадковості.

Програма повинна реалізувати 5 різних способів генерації випадкових чисел. Оцінка випадковості виконується або теоретично, або шляхом статистичної обробки (знаходження математичного сподівання, дисперсії, моди, медіани) та кількості повторюваних чисел у достатньо великій послідовності. Імовірність повтору-відношення кількості повторів на кількість чисел. За необхідності викладач надає спеціаліста, який пояснює статистичної обробки та випадкових чисел. Студент реалізує програму генератора випадкових чисел з оцінкою їх якості.

Вхідні дані: ступінь випадковості, статистичні параметри, кількість випадкових чисел

Вихідні дані: послідовність випадкових чисел, оцінка випадковості цієї послідовності.

Складність: середня.

Примітки: 3, М, С.

4. Реалізація програми, яка перевіряє цифровий підпис.

Реалізувати програму, яка буде перевіряти наявність цифрового підпису у деякої програми, його дійсність та відповідність реальному, і на основі цих даних робитиме висновок про ступінь довіри до цієї програми(довірена, підозріла, недовірена) та пояснюватиме причину.

Вхідні дані: програма, яка потребує перевірки цифрового підпису.

Вихідні дані: стан цифрового підпису і оцінка довіри програмі.

Складність: середня.

Примітки: 3.

5. Реалізація шифру Віженера та простої підстановки.

Програма повинна реалізувати шифрування тексту довільної довжини шифром Віженера та простої підстановки. Шифр повинен мати крім достатньої стійкості також додаткові способи захисту(наприклад-додавання «криптографічної солі(salt)», перевірка користувача, затримку введення в часі).

Вхідні дані: незашифрований текст.

Вихідні дані: зашифрований текст.

Складність: низька.

Примітки: 3.

6. Реалізація геометричного шифру.

Програма повинна реалізувати шифрування тексту довільної довжини геометричним шифром. Літери повідомлення вписуються у вигляді шестикутника, а записуються у порядку ейлерового шляху. Дешифрування відбувається навпаки.

Шифр повинен мати також додаткові способи захисту(наприклад-додавання «криптографічної солі(salt)», перевірка користувача, затримку введення в часі).

Вхідні дані: незашифрований текст.

Вихідні дані: зашифрований текст.

Складність: середня.

Примітки: 3, М.

Різні задачі

7. Реалізація програми для балістичної експертизи.

Балістична експертиза полягає у знаходженні місця, напряму, кута, початкової швидкості руху твердого предмета(наприклад кулі) по відомому куту та глибині проникнення, масі цього предмета у матеріалі(механічна міцність, в'язкість цього матеріалу також відомі). Програма повинна за заданими вхідними даними визначити всі початкові дані. Відтворення польоту тіла за відомими даними необхідно показати графічно.

Вхідні дані: глибина проникнення, маса предмета, матеріал(механічна міцність, в'язкість).

Вихідні дані: місце, напрям, кут, початкова швидкість руху предмета.

Складність: середня.

Примітки: М, С.

8. Реалізація програми для відображення дерева версій Unix з можливістю додавання нових версій.

Програма повинна показати схематично повністю або частково гілки розвитку системи Unix із класифікацією за поколінням, системою та роком випуску. Програма за бажанням користувача повинна показати частину всього дерева із версіями, створеними за вказаний період. Повинна бути можливість додавання нових версій користувачем.

Вхідні дані: існуючі версії Unix, нові версії Unix.

Вихідні дані: дерево версій Unix.

Складність: середня.

Примітки: Г.

9. Реалізація запису структурованого тексту в базу даних з можливістю підтримки NoSQL.

Програма повинна записувати із файлу текст, який має певну структуру(список) зі збереженням цієї структури у базу даних типу не SQL (або SQL-подібну), і при потребі витягати текст із цієї бази даних у новий текстовий файл зі збереженням структури. База даних та технологія запису у неї повинна уточнюватись на етапі технічного завдання. Дозволяється використати відкритий код Jasper Report з проекту Jasper Studio.

Вхідні дані: структурований текст у файлі.

Вихідні дані: запис у базі даних типу NoSQL.

Складність: висока.

Примітки: Б, Т.

10. Реалізація для Linux Ubuntu запису форматowanego тексту в базу даних.

Програма повинна записувати із файлу текст зі збереженням форматування у базу даних, і при потребі витягати текст із цієї бази даних у новий текстовий файл зі збереженням форматування. Дозволяється використовувати не лише Ubuntu, а й іншу систему Linux/ Unix. Якщо даної ОС немає у розробника, для установки емулятора можна використати програму VirtualBox, всередині якої буде запущена сама програма.

Вхідні дані: форматований текст у файлі.

Вихідні дані: запис у базі даних.

Складність: середня.

Примітки: Б, Т.

11. Реалізація будильника з використанням JBoss Seam.

Програма повинна реалізувати будильник, який видаватиме звуковий сигнал(можна разом із графічним), поки користувач не скасує цей сигнал, або не пройде 5 хвилин від старту будильника. Дія вимкнення будильника має бути не дуже очевидна, щоб її не можна було миттєво зрозуміти, як вимкнути-наприклад через математичний приклад. Користувач повинен мати можливість поставити затримку на 5 хвилин, змінити звуковий сигнал на свій та періодично його змінювати. Обов'язкове використання для реалізації програми каркасу JBoss Seam.

Вхідні дані: час запуску будильника, звуковий файл користувача.

Складність: висока.

Примітки: Т.

12. Реалізація програми для розрахунку та відображення поточної фази місяця.

Програма повинна реалізовувати розрахунок фази місяця залежно від заданої дати та часу і зображувати графічно. Також у програму може бути додана опція врахування географічного положення.

Вхідні дані: бажана дата та час.

Вихідні дані: рисунок фази місяця.

Складність: середня.

Примітки: М, Г.

13. Реалізація мобільної програми для виведення інформації про речовину за введеним Е-кодом.

Програма повинна реалізовувати на мобільному пристрої пошук харчової добавки за введеним Е-кодом та виводити інформацію про неї: назва речовини, ступінь шкідливості, чи є заборонена/не рекомендована в Україні, США, ЄС, галузь використання, призначення, основні властивості(якщо є).

Вхідні дані: номер Е-коду.

Вихідні дані: речовина, ступінь шкідливості, чи є заборонена/не рекомендована, галузь використання, призначення, основні властивості.

Приклад: Е621 - Глутамат натрію однозаміщений, Monosodium Glutamate; підсилювач смаку і аромату; схвалений в ЄС, Україні.

Е330 - Лимонна кислота, Citric acid; додається за технологічною необхідністю, в тому числі в какао, шоколадні вироби; схвалений в ЄС, Україні.

Е1510 - Етиловий спирт; розчинник; схвалений в Україні.

Складність: середня.

Примітки: П.

14. Реалізація програми, яка переводить цифровий код у штрих-код і визначає виробника товару.

Програма повинна реалізовувати(на мобільному пристрої або ПК) за введеним числовим кодом малювання штрих-коду, який можна буде роздрукувати, та виводити інформацію про країну-виробника. Окремо програма повинна повідомляти користувача, у випадку якщо даний товар вироблено в Україні та застерігати, якщо товар вироблено на росії.

Вхідні дані: номер штрих-коду.

Вихідні дані: рисунок штрих-коду, країна-виробник.

Складність: низька.

15. Реалізація програми, яка переводить цифровий код у QR-код і визначає виробника товару.

Програма повинна реалізовувати(на мобільному пристрої або ПК) за введеним числовим кодом(або іншим текстом) малювання QR-коду, який можна буде роздрукувати.

Вхідні дані: число/текст.

Вихідні дані: рисунок QR -коду.

Складність: середня.

16. Реалізація програми для малювання української вишивки, характерної для Вінниччини.

Програма повинна виконувати за введеними розмірами(у точках) малювання української вишивки, враховуючи симетрію, у вигляді прямокутного зображення або у вигляді рамки(всередині будь-яке зображення, наприклад логотип фірми). Дозволяється відхилятися від традиційного орнаменту та використовувати орнаменти будь-яких регіонів. Залежності від конкретизованого технічного завдання, складність може варіюватися від низької до високої.

Вхідні дані: розмір рисунка.

Вихідні дані: рисунок вишивки.

Складність: висока.

Примітки: Г.

17. Реалізація програми для малювання української вишивки за способом «Бродівське письмо».

Програма повинна виконувати за введеним текстом малювання української вишивки за способом «Бродівське письмо», враховуючи симетрію. Зображення може бути у вигляді кільцевого надпису, у вигляді простого тексту, так і у вигляді рамки(всередині будь-яке зображення, наприклад логотип фірми). Можна використовувати як прямі, так і косі літери. Дозволяється відхилятися від традиційного орнаменту. Детальний опис способу вишивки «бродівське письмо» описаний у книзі Підгірняка В.П. «Текстова вишивка. Бродівське письмо», 2008 року. Залежності від конкретизованого технічного завдання, складність може варіюватися від низької до високої.

Вхідні дані: розмір рисунка.

Вихідні дані: рисунок вишивки.

Складність: висока.

Примітки: Г.

Мобільні пристрої

18. Реалізація будильника для мобільного пристрою.

Програма повинна реалізовувати будильник для мобільного пристрою(телефону, планшета), який видаватиме звуковий сигнал(можна разом із графічним), поки користувач не скасує цей сигнал, або не пройде 5 хвилин від старту будильника. Дія вимкнення будильника має бути не дуже очевидна, щоб її не можна було миттєво зрозуміти, як вимкнути-наприклад повторення руху на екрані. Користувач повинен мати можливість поставити затримку на 5 хвилин, змінити звуковий сигнал на свій та періодично його змінювати.

Вхідні дані: час запуску будильника, звуковий файл користувача.

Складність: середня.

Примітки: П.

19. Реалізація програми для упорядковувала музики для мобільного пристрою.

Програма повинна реалізовувати для мобільного пристрою(телефону, планшета) створення таблиці всіх наявних на пристрої музичних файлів, де

вказуватиме їх місцезнаходження, розмір, дату створення. Також програма дозволяє шукати у ній, сортувати та запускати ці файли, рахує кількість запусків кожного файлу через програму, дає можливість додавати автора, жанр, групу, виконавця композиції.

Вхідні дані: музичні файли на мобільному пристрої.

Вихідні дані: впорядкована таблиця музичні файли на мобільному пристрої.

Складність: висока.

Примітки: П.

20. Реалізація простого диспетчера задач для мобільного пристрою

Програма повинна реалізовувати для мобільного пристрою(телефону, планшета) створення таблиці всіх запущених на пристрої програм, відображати запущені процеси та потоки, а також вказувати їх місцезнаходження, розмір використовуваної пам'яті, час запуску. У програмі повинна бути можливість запускати свій процес (запустити програму від користувача).

Вхідні дані: запущені програми на мобільному пристрої.

Вихідні дані: впорядкована таблиця запущених програми на мобільному пристрої.

Складність: висока.

Примітки: П.

21. Реалізація програми для визначення поточного місцезнаходження для мобільного пристрою.

Програма повинна реалізовувати для мобільного пристрою(телефону, планшета) знаходження користувача у поточний момент, відобразити його положення на карті, вивести його координати та при потребі відправити їх через повідомлення/СМС/електронну пошту. Програма повинна однаково відобразитись на планшеті та мобільному телефоні. Можна використати взаємодію з картами Google.

Вихідні дані: поточні координати користувача та його положення на карті.

Складність: низька.

Примітки: П.

22. Реалізація програми для визначення довжини маршруту від поточного місцезнаходження для мобільного пристрою.

Програма повинна реалізовувати для мобільного пристрою(телефону, планшета) знаходження маршруту для користувача за відомими координатами або точкою на карті початку і кінця маршруту. Програма повинна відобразити маршрут на карті і розрахувати його довжину. Можна використати взаємодію з картами Google.

Вхідні дані: початкові і кінцеві координати маршруту користувача і їх положення на карті.

Вихідні дані: маршрут на карті та його довжина.

Складність: середня.

Примітки: П.

Соцмережі

23. Реалізація автоматичного відповідача у нічний час для соціальної мережі Facebook.

Програма повинна реалізовувати для соціальної мережі Facebook автоматичний відповідач, який буде відписувати (без втручання користувача сторінки) всім користувачам певний текст (наприклад «користувач на даний момент спить, напишіть йому вранці») у нічний час(з 0:00 до 7:00). Даний текст задається в програмі. У програмі потрібно передбачити кілька варіантів тексту, які вибираються залежно від певних умов(часу написання, номеру повідомлення, людини, яка пише, або категорії дописувачів). Причому ця програма запам'ятовує користувача, який писав вночі, та у час, коли в нього буде поздоровлення, привітає також вночі. У програмі потрібно передбачити кілька варіантів тексту поздоровлення, які вибираються залежно від певних умов(дати та часу поздоровлення, людини, яку вітають, або її категорії).

Вхідні дані: час та текст відповіді.

Вихідні дані: повідомлення, написані іншими користувачами вночі.

Складність: середня.

24. Реалізація програми для автоматичного написання відтермінованих повідомлень у заданий час для соціальної мережі Twitter.

Програма повинна реалізовувати для соціальної мережі Twitter автоматичний відповідач, який буде писати певні тексти («твіти») (без втручання користувача сторінки) у заданий час всім підписаним користувачам, або заданим користувачам чи категоріям користувачів. Даний текст(з #хештегами) задається в програмі. У програмі потрібно передбачити кілька варіантів тексту, які вибираються залежно від певних умов(часу викладення, певного повідомлення іншого користувача, номеру повідомлення).

Вхідні дані: час та текст повідомлень.

Вихідні дані: коментарі, кількість лайків та репостів кожного повідомлення.

Складність: середня.

25. Реалізація програми для автоматичного виставлення фотографій у заданий час для соціальної мережі Instagram.

Програма повинна реалізовувати для соціальної мережі Instagram автоматичне виставлення фотографій (без втручання користувача сторінки) у заданий час. Зображення з текстовим підписом(за бажанням з #хештегами, геомітками) задається в програмі у вигляді файлу або шляху до нього. У програмі потрібно передбачити можливість виставлення фотографії залежно від певних умов(часу викладення, певного повідомлення іншого користувача, номеру фотографії, дати, поточного перебування користувача, певних коментарів інших користувачів).

Вхідні дані: час, дата і файли фотографій для виставлення, або шлях на ПК до них, текст надпису до них.

Вихідні дані: коментарі, кількість лайків та репостів кожної фотографії.

Складність: середня.

Інтернет, локальна мережа

26. Реалізація автоматизованого перекладача з використанням Hibernate.

Програма повинна перекладати у автоматичному режимі текст, який вписаний у дану програму, або перекладати вказаний текстовий файл. Переклад можна виконувати будь-яким способом. Обов'язкове використання каркасу(фреймворка) Hibernate(будь-якої її частини) або аналогічного.

Вхідні дані: Текст для перекладу/файл для перекладу.

Вихідні дані: Перекладений текст/перекладений файл.

Складність: низька.

Примітки: Т.

27. Реалізація програми для відстежування курсу валют із графічним відображенням коливань на основі Spring Web Flow (SWF).

Програма повинна у автоматичному режимі відстежувати зміни курсу вказаних валют протягом певного часу і оновлювати дані. У якості джерела інформації можна використати офіційний сайт НБУ. Користувач повинен мати можливість побачити коливання однієї або кількох валют протягом певного часу у вигляді графіка. Обов'язкове використання каркасу(фреймворка) Spring Web Flow (будь-якої її частини) або аналогічного.

Вхідні дані: Значення курсу валют відносно гривні, періодичність оновлення, період коливань.

Вихідні дані: Поточний курс вказаних валют, значення коливань за вказаний період.

Складність: низька.

Примітки: Т.

28. Реалізація програми для автоматичної відправки повідомлень по електронній пошті у вказаний час за певних умов.

Програма повинна у вказаний час у автоматичному режимі (без участі користувача) відправляти повідомлення з підготовленим текстом за вказаною адресою. Також можуть бути додаткові умови для часу та адреси відправки повідомлення(зміст тексту, значення ціни на нафту у даний час, перемога чи поразка деякої футбольної команди). Ці додаткові умови повинен або створюватись користувачем(для цього треба передбачити відповідний інтерфейс), або отримувати у вигляді програмного модуля, або отримувати у вигляді значення у текстовому файлі(яке може змінюватись, і залежить від цих умов).

Вхідні дані: Текст повідомлення для відправки, адресат, відправник, час відправлення.

Вихідні дані: Повідомлення про стан виконання.

Складність: середня.

29. Реалізація власного простого завантажувача сайту для автономного перегляду при відсутності доступу до мережі інтернет.

Програма повинна виконувати завантаження цілого сайту у вказану папку, при необхідності змінювати посилання всередині. Сайт повинен у автономному режимі зберігати зображення, відео, вбудовані об'єкти. Повинні працювати внутрішні посилання, при необхідності змінити посилання відносно папки завантаження як кореневої папки.

Вхідні дані: IP-адреса/ім'я хоста сайту, тип програми-клієнта(її номер порта), папка для збереження.

Вихідні дані: Час збереження, збережений сайт.

Складність: середня.

30. Реалізація програми для збереження зображення з веб-сайту на комп'ютер.

Програма повинна зберігати зображення з вказаного веб-сайту на жорсткий диск комп'ютера користувача за вказаною адресою. Користувач повинен мати можливість обмежити розмір, якість (кількість пікселів) та тип зображень для збереження. Про стан завантаження(успішно/неуспішно) формується звіт у вигляді списку із вказуванням параметрів кожного збереженого зображення.

Вхідні дані: Посилання на сайт, список типів зображень для збереження, максимальний і мінімальний розмір і якість зображень, папка для збереження.

Вихідні дані: Кількість збережених зображень, список, розмір і тип збережених зображень, збережені зображення.

Складність: середня.

Примітки: Г.

Реалізація програми для обміну повідомленнями (месенджера):

31. Частина 1. Реалізація інтерфейсу користувача(GUI).

Програма повинна реалізовувати графічний інтерфейс користувача(GUI), який надає можливість введення користувача, його авторизація(перевірка паролю користувача, завантаження його даних і статистики), а також інтерфейс(форми для введення даних) для під'єднання по локальній мережі. Також програма повинна реалізувати графічно панель меню, відображає кнопки в меню, відображає прийнятий та відправлений текст(взаємодія із модулями частини 2-4, графічний формат та використовувані технології визначає розробник цієї частини).

Вхідні дані: ім'я користувача, пароль, адреса в мережі(ім'я хоста, IP-адреса), номер порта, текст повідомлень для відправки супернику.

Складність: середня.

Примітки: К, Г.

32. Частина 2. Реалізація обміну текстовими повідомленнями через інтернет та локальну мережу.

Програма повинна реалізовувати можливість доставки текстових повідомлень віддаленому користувачу через локальну (LAN) або глобальну (Internet) мережу, використовуючи введені користувачем дані з'єднання (отримані від модуля частини 1). Отриманий та відправлений текст передається для відображення модулю частини 2. Програма повинна відслідковувати час отримання повідомлення і видати повідомлення у випадку неможливості спілкування з іншим користувачем. Також повинна бути можливість відправити текст із підготовленого текстового файлу.

Вхідні дані: Текст для відправлення, час відправлення.

Вихідні дані: Отриманий текст, час отримання, стан доставки повідомлення.

Складність: середня.

Примітки: К.

33. Частина 3. Реалізація обміну файлами через інтернет та локальну мережу.

Програма повинна реалізувати можливість доставки файлових повідомлень віддаленому користувачу через локальну (LAN) або глобальну (Internet) мережу, використовуючи введені користувачем дані з'єднання (отримані від модуля частини 1). При успішному отриманні та відправленні файлу відображається відповідне повідомлення. Програма повинна відслідковувати час отримання файлу, показувати, яку частину даних уже отримано і видати повідомлення у випадку неможливості спілкування з іншим користувачем.

Вхідні дані: Файл для відправлення, адреса розміщення файлу час відправлення.

Вихідні дані: Отриманий файл, час отримання, стан доставки повідомлення.

Складність: висока.

Примітки: К.

34. Частина 4. Реалізація збереження історії повідомлень у базі даних із можливістю пошуку.

Програма повинна реалізувати можливість збереження переписки(в т.ч. час доставки і отримання кожного повідомлення та файлу) користувача за даний сеанс і дописувати у базу даних до попередньої історії, фіксуючи дату, час початку і кінця спілкування, кількість повідомлень, список отриманих файлів. Програма повинна надавати можливість пошуку у історії у базі даних за фрагментом тексту, знаходити текст у вказаний період(задається час початку і кінця).

Вхідні дані: Текст (із часом) переписки під час сеансу, попередня історія, час початку і кінця для пошуку, фрагмент тексту для пошуку.

Вихідні дані: Текст історії, знайдений текст за вказаний період, знайдений текст.

Складність: середня.

Примітки: К, Б.

Реалізація власної простої програми для доставки файлів за технологією торент-трекер:

35. Частина 1. Реалізація інтерфейсу користувача(GUI).

Програма повинна реалізувати графічний інтерфейс користувача(GUI), надавати поля введення даних IP-адреси і порт у режимі «ручного» підключення, надавати меню для всіх операцій, необхідних розробникам частин 2-6, надати відображення графіку(або діаграми) зміни швидкості з'єднання, часу передавання за отриманою послідовністю даних, відображати кількість переданих пакетів від кожного сервера та кількість переданих пакетів кожному клієнту.

Вхідні дані: IP-адреси і порт підключення програм торент-клієнтів.

Вихідні дані: IP-адреси і порт підключення програм торент-клієнтів.

Складність: середня.

Примітки: К, Г.

36. Частина 2. Реалізація простого torrent-клієнта.

Програма повинна використовувати технологію «торент-трекер» в режимі клієнта, тобто мати можливість отримувати файли частинами від віддаленої програми, або яка працює торент-сервером(роздає даний файл, сервер може бути не один). Технологію «торент-трекер» можна використовувати будь-якої версії, або реалізувати власну, зберігши дотримання лише загальних принципів(для програми важлива працездатність, а не відповідність технології). Також у програмі повинні

показуватись кількість отриманих пакетів(або фрагментів файлів), графік швидкості зачачки а після зачачки середню швидкість і час зачачки.

Вхідні дані: Торент-файл(торент-посилання для під'єднання до серверів), адреса для збереженні на комп'ютері, IP-адреса і порт підключення програм торент-серверів.

Вихідні дані: Час зачачування файлу, кількість фрагментів файлу, середня швидкість зачачки файлу, графік швидкості зачачки, ступінь(%) зачачки файлу в кожний момент часу.

Складність: середня.

Примітки: К.

37. Частина 3. Реалізація простого torrent-сервера(для роздачі).

Програма повинна використовувати технологію «торент-трекер» в режимі сервера, тобто мати можливість віддавали файли частинами будь-якій віддаленій програмі, або яка працює торент-клієнтом(отримує даний файл, клієнт може бути не один) і відправила запит на отримання файлу. Технологію «торент-трекер» можна використовувати будь-якої версії, або реалізувати власну, зберігши дотримання лише загальних принципів (для програми важлива працездатність, а не відповідність технології). Також у програмі повинні показуватись кількість переданих пакетів (або фрагментів файлів), графік швидкості викачки а після викачки середню швидкість і час зачачки, частка втрачених пакетів і переданих повторно.

Вхідні дані: Торент-файл, адреса файла на комп'ютері, IP-адреси і порт підключення програм торент-клієнтів.

Вихідні дані: Час викачування файлу, кількість фрагментів файлу, середня швидкість викачки файлу, графік швидкості викачки, ступінь(%) викачки файлу в кожний момент часу, частка(%) втрачених пакетів і переданих повторно.

Складність: середня.

Примітки: К.

38. Частина 4. Реалізація програми для оптимізації швидкості при нестабільній та різній швидкості з'єднання у користувачів.

Програма повинна виконувати оптимізацію ресурсу інтернет-каналу у програми сервера, якщо швидкість під'єднання у кожного користувача різна і може змінюватись у часі. Програма повинна показувати зі зміною у часі розподіл ресурсу між користувачами. Надавати ресурс потрібно пропорційно можливостям клієнта. Для вирішення даної задачі можна використати розділ теорії оптимізації «системи масового обслуговування».

Вхідні дані: Торент-файл, адреса файла на комп'ютері, IP-адреси і порт підключення програм торент-клієнтів, середня та зміна в часі швидкості з'єднання кожного користувача.

Вихідні дані: Частка ресурсу інтернет-каналу, яка виділяється кожному користувачу у даний момент часу. Час викачування файлу, кількість фрагментів файлу, середня швидкість викачки файлу, графік швидкості викачки, ступінь(%) викачки файлу в кожний момент часу, частка(%) втрачених пакетів і переданих повторно.

Складність: висока.

Примітки: К, М.

39. Частина 5. Реалізація програми для оптимізації часу доставки та маршруту проходження пакетів.

Програма повинна виконувати оптимізацію ресурсу інтернет-каналу у програмі сервера, розраховуючи, що швидкість під'єднання у кожного користувача більш-менш стабільна. Оптимізація проводиться шляхом вирішення задачі маршрутизації(зазвичай цю задачу вирішує відповідне обладнання маршрутизаторів) пошуку інших, більш коротких(у часі) шляхів доставки пакетів. При знаходженні коротшого шляху доставка пакетів відбувається по цьому маршруту. Програма повинна показувати зміну часу доставки пакетів та шлях(повний зі список усіх проміжних точок, або скорочений) проходження пакетів. Для вирішення даної задачі можна використати будь-який розділ теорії оптимізації, який виконує знаходження оптимального маршруту між двома точками(пошук в ширину, пошук в глибину, жадібний, швидкий алгоритми...).

Вхідні дані: Торент-файл, адреса файла на комп'ютері, IP-адреси і порт підключення програм торент-клієнтів.

Вихідні дані: Повний(або скорочений) шлях проходження пакетів під час доставки із вказуванням усіх проміжних пристроїв маршрутизації. Час викачування файлу, кількість фрагментів файлу, середня швидкість викачки файлу, графік швидкості викачки, ступінь(%) викачки файлу в кожний момент часу, частка(%) втрачених пакетів і переданих повторно.

Складність: висока.

Примітки: К, М.

40. Частина 6. Реалізація програми для захищеного передавання (криптографічне шифрування пакетів та аутентифікація користувачів).

Програма повинна виконувати аутентифікацію користувачів на клієнтській частині(перевірка імені та хешованого пароллю користувача на серверній частині, та виводити повідомлення про стан аутентифікації-успішно, неуспішно, багатократно неуспішно) та виконувати криптографічне шифрування пакетів з відкритим ключем на серверній частині перед відправленням на розшифрування пакетів у клієнтській програмі. Обрахувати зменшення швидкості передавання за рахунок шифрування та час, витрачений на шифрування та дешифрування.

Вхідні дані: Торент-файл, адреса файла на комп'ютері, IP-адреси і порт підключення програм торент-клієнтів, ім'я і пароль користувача, відкритий ключ шифрування.

Вихідні дані: Зашифровані пакети(або фрагменти файлу) для відправлення, стан аутентифікації користувача.

Складність: середня.

Примітки: К, З.

Веб-технології

41. Створення інформаційного веб-сайту на HTML5.

Необхідно написати інформаційний веб-сайт (портал) з динамічним контекстом(відео, анімація, скрипти, аплети) на основі HTML5. На самому сайті повинні бути посилання на інші сторінки порталу, а також посилання на сторінки «карту сайта», «головна», «детальна інформація», «про авторів». Необхідно зберігати IP-адреси всіх відвідувачів(можна із класифікацією за регіоном).

Обовязкове тестування розміщення сайту на власному сервері(можна взяти Denwer, Tomcat, GlassFish або безкоштовні онлайн-хостинги)

Вхідні дані: Контент сайту

Вихідні дані: IP-адреси відвідувачів сайту.

Складність: середня.

Примітки: В, Т.

42. Створення інформаційного веб-сайту з можливістю перегляду фотографій на DHTML.

Необхідно написати інформаційний веб-сайт (портал) з динамічним контекстом (перегляд багатьох фотографій) на основі DHTML. Повинна бути можливість додавання фотографій певним категоріям привілейованих користувачів. На самому сайті повинні бути посилання на інші сторінки порталу, а також посилання на сторінки «карту сайта», «головна», «детальна інформація», «гостьова сторінка», «про авторів».

Вхідні дані: Контент сайту, фотографії, логін та пароль користувача

Вихідні дані: Категорія користувача

Складність: середня.

Примітки: В, Т.

43. Створення веб-сайту-форуму із БД DB/2 на HTML не нижче 4.0.

Необхідно написати веб-сайт (форум), з динамічним контекстом (перегляд зображень користувача та скачування доданих файлів) на HTML 4.0 і вище. Повинна бути можливість додавання фотографій та файлів(дозволених типів) певним категоріям користувачів(які написали певну кількість повідомлень за певний час). На форумі повинен бути список тем для обговорень, а обговорення у певній темі відображається у вигляді вітки повідомлень. Відповідь одного користувача на повідомлення іншого позначається у вигляді вітки та іншим кольором. На сайті існує ієрархія зареєстрованих користувачів. На самому сайті повинні бути посилання на інші сторінки порталу, а також посилання на сторінки «карту сайта», «головна», «список тем», «гостьова сторінка», «авторизація», «особистий кабінет», «про авторів». Весь текст обговорення та файли зберігаються у базі даних DB/2.

Вхідні дані: Контент сайту, фотографії, прикріплені файли, список користувачів, логіни та паролі користувачів.

Вихідні дані: Текст усіх тем у БД.

Складність: середня.

Примітки: В, Т, Б.

44. Створення інформаційного веб-сайту, який здатен працювати з БД.

Необхідно написати інформаційний веб-сайт (спеціалізований новинний портал) з динамічним контекстом(зображення, відео, анімація, скрипти, аплети), який надає можливість певним категоріям користувачів додавати інформацію у вигляді стрічки новин RSS. Користувачам-гостям доступні не всі новини, авторизованим користувачам-майже всі, авторам, адміністраторам, VIP-користувачам-усі. На самому сайті повинні бути посилання на інші сторінки порталу, а також посилання на сторінки «карту сайта», «головна», «список тем», «гостьова сторінка»,

«авторизація», «особистий кабінет», «новини», «про авторів». Для робіт із БД можна використати технології JSP або Servlet.

Вхідні дані: Контент сайту, текст усіх новин, текст новини для додання.

Вихідні дані: Текст усіх новин, користувачі у БД, категорія користувача.

Складність: низька.

Примітки: В, Б.

45. Створення інформаційного веб-сайту із можливістю перегляду відео на JavaFX.

Необхідно написати інформаційний веб-сайт (портал збірник блогів) з динамічним контекстом (перегляд відео) на основі HTML та з використанням JavaFX. Відео може бути вставлене з відеосерверів(YouTube) або бути доданим користувачем. Повинна бути можливість вибрати кращу або гіршу якість відео(за наявності обох видів). Має бути можливість додавання відео з описом певним категоріям користувачів. На самому сайті повинні бути посилання на інші сторінки користувачів, а також посилання на сторінки « карту сайта», «головна», «пошук відео по темі», «гостьова сторінка», «про авторів».

Вхідні дані: Контент сайту, відео, якість і опис відео, логін та пароль користувача

Вихідні дані: Категорія користувача, доступні відео для певного користувача.

Складність: висока.

Примітки: В, Т.

46. Створення веб-сайту для музиканта з можливістю керувати доступом відтворенням композицій залежно від рейтингу користувача на сайті.

Необхідно написати інформаційний веб-сайт (музичний портал-блог) на з динамічним контекстом (прослуховування музичних файлів). Звукові файли можуть бути вставлене з відповідних сервісів(SoundCloud) або бути доданим користувачем. Кожен звуковий файл має відомості: якість, формат файлу, автор, виконавець, дата запису, жанр, група-виконавець, альбом, тип(музика, голос, професійна музика, мелодія) та опис. Має бути можливість додавання звукових файлів з описом певним категоріям користувачів та прослуховування звукових файлів залежно від рейтингу користувача(кількість авторських композицій, якість-оцінка від інших користувачів). На самому сайті повинні бути посилання на інші сторінки користувачів, а також посилання на сторінки « карту сайта», «головна», «доступні композиції», «мої композиції», «пошук композиції»(по жанру, виконавцю, автору, альбому...), «гостьова сторінка», «авторизація», «про авторів».

Вхідні дані: Контент сайту, аудіо-файли, відомості про них, логін та пароль користувача

Вихідні дані: Категорія користувача, рейтинг користувача, доступні аудіо для певного користувача.

Складність: середня.

Примітки: В.

47. Створення веб-сайту для перегляду інформації про певний товар зі списку із використанням JavaServer Faces (JSF).

Необхідно написати інформаційний веб-сайт, який надає інформацію про певні товари з інтернет-магазину з динамічним контекстом (зображення, відео, скрипти). Зображення та відео, та інформацію про товар можуть додавати лише

адміністратор. Користувач запитує товар за ключовими словами, пошук яких виконується у базі даних серверною частиною за технологією JSF. На самому сайті повинні бути посилання на сторінки «карту сайта», «головна», «доступні товари», «пошук товару», «про нас».

Вхідні дані: Контент сайту, список товарів, опис всіх товарів, текст пошуку.

Вихідні дані: Знайдений результат.

Складність: середня.

Примітки: В, Т, Б.

48. Створення веб-сайту, який доступний на відкритому сервері та зберігає інформацію, захищену від хакерських атак (будь-хто зі групи має можливість спробувати викрасти цю інформацію).

Необхідно написати веб-сайт, який розміщений на одному з безкоштовних хостів(або хост створений на власному комп'ютері), зберігає деяку інформацію, доступну для деяких користувачів, мають доступ. Кількість доступів, користувач і час доступу суворо фіксуються у журналі. Інформація має бути захищена від хакерських атак(підбір пароля, переповнення стека, обхід доступу через помилки пам'яті, ...). Будь-яка людина може спробувати викрасти цю інформацію(і на сайті про це написано). Якщо інформацію викрав студент(и), він отримує(ють) підвищення оцінки з предмету «програмування» на 10 балів. Якщо інформація була викрадена, розробник повинен внести вдосконалення і закрити вразливість системи. Атаки вважаються витримані, якщо на протязі тижня інформація не була викрадена.

Вхідні дані: Інформація, користувачі, які мають доступ.

Вихідні дані: Кількість доступів, користувач і час доступу.

Складність: середня.

Примітки: В, З.

49. Створення веб-сайту для масової гри у «хрестики-нулики» онлайн на скриптах(PHP/JavaScript/JScript/Node JS/Ruby/Python/JQuery/Elixir/ін.).

Необхідно написати веб-сайт для масової гри (багатьма користувачами паралельно) у «хрестики-нулики» (розмір 100 x 100) по парам онлайн, який використовує скрипти (PHP/JavaScript/JScript/Node JS/Ruby/Python/JQuery/Elixir/ін.). Користувачі мають можливість авторизуватись або грати анонімно. У чемпіонаті приймають участь лише зареєстровані користувачі. Сайт реєструє та відображає кількість перемог користувача, час гри, 5 кращих гравців за тиждень.

Вхідні дані: Ім'я користувача і пароль.

Вихідні дані: Кількість перемог користувача, час гри, 5 кращих гравців за тиждень.

Складність: середня.

Примітки: В, Т.

50. Створення веб-сайту для масової гри у «шашки» онлайн із БД на Google Web Toolkit.

Необхідно написати веб-сайт для масової гри (багатьма користувачами паралельно) у «шашки» по парам онлайн. Сама повинна використовувати Google Web Toolkit. Користувачі мають можливість авторизуватись або грати анонімно. У чемпіонаті приймають участь лише зареєстровані користувачі. Сайт реєструє та відображає кількість перемог користувача, час гри, 5 кращих гравців за тиждень.

Вхідні дані: Ім'я користувача і пароль.

Вихідні дані: Кількість перемог користувача, час гри, 5 кращих гравців за тиждень.

Складність: середня.

Примітки: В, Т.

Створення веб-сайту для масової гри у «морський бій» онлайн, який буде надавати можливість грати багатьом користувачам окремо та між собою.

51. Частина 1. Створення веб-сайту із авторизацією користувачів.

Необхідно написати веб-сайт для масової гри (багатьма користувачами паралельно) у «морський бій» по парам онлайн. У даній частині сайту відбувається авторизація користувача, перевірка хешованого пароля, захист через введення текст з картинок(CARTNA), можливість зареєструватись новому користувачу і відновити пароль. На сайті суперник вибирається автоматично(той, хто зайшов відразу після даного користувача, який шукає пару). Обов'язкове використання Unity 3D.

Вхідні дані: Ім'я користувача і пароль, зареєстровані онлайн користувачі.

Вихідні дані: Результат правильності введення.

Складність: низька.

Примітки: В, Т.

52. Частина 2. Створення графічного інтерфейсу гри у «морський бій»

Необхідно написати веб-сайт для масової гри (багатьма користувачами паралельно) у «морський бій» по парам онлайн. У даній частині сайту виконується надання графічного інтерфейсу, меню, підготовка поля для розстановки кораблів, відображується процес битви і виведення повідомлення про перемогу чи поразку відповідному користувачу. Обов'язкове використання Unity 3D.

Вхідні дані: Суперники(пара користувачів), натискання стрілок для розстановки кораблів, координати пострілів.

Вихідні дані: Координати цілих та потоплених(підбитих) кораблів, переможець.

Складність: середня.

Примітки: В, Г, Т.

53. Частина 3. Створення штучного інтелекту для гри у «морський бій».

Необхідно написати веб-сайт для масової гри (багатьма користувачами паралельно) у «морський бій» по парам онлайн. У даній частині сайту виконується надання можливості гри без іншого користувача(зі штучним інтелектом) з використання графічного інтерфейсу від розробника частини 2. Необхідно зробити 3 рівні складності штучного інтелекту. Відображується процес битви і виведення повідомлення про перемогу чи поразку відповідному користувачу. Обов'язкове використання Unity 3D.

Вхідні дані: Натискання стрілок для розстановки кораблів, координати пострілів.

Вихідні дані: Координати цілих та потоплених(підбитих) кораблів, переможець.

Складність: висока.

Примітки: В, М, Т.

54. Частина 4. Створення інтерфейсу та функцій для ведення статистики на сайті та записом у БД.

Необхідно написати веб-сайт для масової гри (багатьма користувачами паралельно) у «морський бій» по парам онлайн. У даній частині сайту виконується створення інтерфейсу та функцій для ведення статистики на сайті за кожним користувачем та всіма користувачами разом, визначення кращих гравців, відбір у таблиці чемпіонату(турнір на протязі дня). Вся статистика записується у базу даних. Кожен користувач бачить власну статистику та статистику сайту в цілому.

Вхідні дані: Ім'я користувача і пароль, зареєстровані онлайн користувачі, їх попередні успіхи.

Вихідні дані: Кількість перемог і поразок, середній час боїв, успішність(%) кожного користувача, 10 кращих гравців, ступінь у чемпіонаті.

Складність: середня.

Примітки: В, Б.

Ігри

Реалізація гри «морський бій» із автоматичним встановленням програми(інсталятор)

55. Частина 1. Реалізація інтерфейсу із авторизацією користувачів та можливістю обміну повідомленнями по локальній мережі.

Програма повинна реалізовувати програмний інтерфейс, який надає можливість введення користувача, його авторизація(перевірка паролю користувача, завантаження його даних і статистики-для цього можна використати модулі розробника частини 2, 3), а також під'єднання по локальній мережі або інтернет, вставлення з'єднання, доставка та відображення повідомлень, відправлених іншим користувачем. Програма повинна відправляти інформацію про користувача супернику та отримувати інформацію про суперника, в т.ч. в процесі гри(координати і успішність пострілу).

Вхідні дані: ім'я користувача, пароль, адреса в мережі(ім'я хоста, IP-адреса), номер порта, текст повідомлень для відправки супернику.

Вихідні дані: ім'я суперника, текст повідомлень, отриманих від суперника.

Складність: середня.

Примітки: К.

56. Частина 2. Реалізація графічного інтерфейсу гри у «морський бій».

Програма повинна реалізовувати графічний інтерфейс користувача(GUI), який надає можливість введення користувача, його авторизація(перевірка паролю користувача, завантаження його даних і статистики), а також інтерфейс(форми для введення даних) для під'єднання по локальній мережі. Також програма повинна графічно представляти статистику користувача на основі отриманих даних від модуля з частини 3. Також програма реалізовує графічно відображає власне ігровий процес для «морського бою». Обов'язкове використання кількох власних спрайтів для гри.

Вхідні дані: ім'я користувача, пароль, адреса в мережі(ім'я хоста, IP-адреса), номер порта, текст повідомлень для відправки супернику.

Складність: середня.

Примітки: К, Г.

57. Частина 3. Реалізація штучного інтелекту для гри у «морський бій».

Програма повинна реалізовувати на основі модулів частини 1, 2 штучний інтелект, який дозволить грати користувачу при відсутності іншого гравця(суперника). Штучний інтелект повинен самостійно розставляти кораблі, вибирати стратегію(та змінювати залежно від успішності), координати для власних пострілів. Стратегії поведінки та оцінка виграшу описуються у розділі математики «теорія ігор».

Вхідні дані: координат пострілів суперника та успішність власних пострілів.

Вихідні дані: координати розстановки кораблів, координати власних пострілів.

Складність: середня.

Примітки: К, М.

58. Частина 4. Реалізація ведення статистики та запису у БД Cloudscape.

Програма повинна реалізовувати програмний інтерфейс, який надає інформацію(статистику боїв) про користувача(для цього можна використати модулі розробника частини 2), а також про суперника користувача. Також даний модуль формує таблицю кращих гравців за всю історію(«дошка слави»). Після завершення бою програма повинна внести нову інформацію у базу даних та перерахувати дані користувачів. Обов'язкове використання для реалізації програми бази даних Cloudscape (дозволяється змінити, але конкретна БД має бути наперед визначена).

Вхідні дані: результати і дані поточного бою, історія і статистика ігор користувача.

Вихідні дані: успішність, статистика і рейтинг користувача та суперника, «дошка слави» кращих гравців.

Складність: середня.

Примітки: К, Б, Т.

59. Частина 5. Реалізація програми встановлення гри із визначенням платформи та перевірки можливості встановлення гри.

Програма, отримавши завершені модулі від розробників частин 1,2,3,4 повинна реалізовувати визначенням платформи користувача, виконати перевірку можливості встановлення усієї програми гри на комп'ютер користувача, у випадку неможливості зазначити причину та запропонувати шляхи вирішення проблеми. Програма повинна мати механізм додання оновлень та патчів при уже встановленій грі, та здійснювати контроль і сумісність версій. Також повинен бути передбачений контроль умовної ліцензії та отримання від користувача згоди на цю ліцензію.

Вхідні дані: платформа користувача, версія гри.

Вихідні дані: допустимість встановлення на платформа користувача, або причина неможливості встановлення програми та шляхи виправлення.

Складність: середня.

Примітки: К.

60. Реалізація гри «пятнашки» для мобільного пристрою.

Програма повинна реалізовувати для мобільного пристрою(телефона, планшета) гру «пятнашки», реалізувати графічний інтерфейс користувача(GUI), можливість управління клавішами/сенсорним екраном, вітати користувача при виграші та записувати результати(час і кількість виграшів). Обов'язкове використання Unity (або його окремих елементів) не нижче 5-ї версії.

Вихідні дані: час гри

Складність: середня.

Примітки: П, Г, Т.

61. Реалізація гри «хрестики-нулики» для мобільного пристрою на Android.

Програма повинна реалізовувати для мобільного пристрою(телефона, планшета) гру «хрестики-нулики», реалізувати графічний інтерфейс користувача(GUI), можливість управління клавішами/сенсорним екраном, вітати користувача при виграші та записувати результати(час і кількість виграшів). Обов'язкове використання OpenGL. Допускається реалізація не лише на Android.

Вихідні дані: час гри

Складність: середня.

Примітки: П, Г, Т.

62. Реалізація гри «монополія»

Програма повинна реалізовувати гру «монополія», реалізувати графічний інтерфейс користувача(GUI), можливість управління клавішами/мишею, передбачити можливість або грати двом гравцям, або одному гравцю зі штучним інтелектом. Програма повинна вітати користувача при виграші та записувати результати(час, параметри і кількість виграшів). Обов'язкове використання Unity 2D.

Вихідні дані: час і параметри гри

Складність: середня.

Примітки: Г, М, Т.

Реалізація програми для гри у пінг-понг по локальній мережі.

63. Частина 1. Реалізація графічного інтерфейсу та штучного інтелекту способом 1.

Програма повинна реалізовувати гру «пінг-понг», реалізувати графічний інтерфейс користувача(GUI) із формами для введення даних для інтернет-з'єднання(використовувати дані форми введення і обробляти дані буде розробник для «частини 2»), а також власне ігрового процесу(відображення гравця, суперника, м'ячика), можливість управління клавішами/мишею, передбачити можливість або грати двом гравцям, або одному гравцю зі штучним інтелектом, а також змагатися даному штучному інтелекту із штучним інтелектом іншого розробника(із частини 2). Програма повинна вітати користувача при виграші та записувати результати(час, параметри і кількість виграшів). Обов'язкове використання WebGL.

Вхідні дані: координати суперника і м'ячика

Вихідні дані: власні координати, час гри, кількість пропущених «м'ячиків» кожного гравця

Складність: середня.

Примітки: М, К, Г.

64. Частина 2. Реалізація з'єднання по Internet та штучного інтелекту способом 2.

Програма повинна реалізовувати програмний інтерфейс, який надає можливість введення користувача, завантаження його даних і статистики, а також під'єднання по локальній мережі або інтернет, вставлення з'єднання, доставка повідомлень із координатами, відправлених іншим користувачем. Також потрібно

реалізувати штучний інтелект, який буде змагатися із штучним інтелектом іншого розробника(з частини 1).

Вхідні дані: ім'я користувача, пароль, адреса в мережі(ім'я хоста, IP-адреса), номер порта, координати суперника і м'ячика.

Вихідні дані: власні координати, час гри, кількість пропущених «м'ячиків» кожного гравця

Складність: середня.

Примітки: М, К.

Реалізація гри «лучник».

65. Частина 1. Реалізація інтерфейсу користувача(GUI).

Програма повинна реалізовувати графічний інтерфейс користувача(GUI), який реалізує графічно власне ігровий процес: фігуру гравця-лучника, який може змінювати силу та кут нахилу; стрілу у польоті(розрахунок координат, траєкторії, вплив вітру виконує розробник частини 2); малювання різних перешкод(властивості перешкод та їх вплив на політ стріли виконує розробник частини 3); надає можливість введення користувача, завантаження його даних і статистики. Обов'язкове використання OpenGL (рекомендується не нижче версії 3.0) або його компонентів SFML/WebGL/ін.

Вхідні дані: ім'я користувача, кількість пройдених рівнів, сила та кут натягу лука, швидкість і напрям вітру.

Складність: висока.

Примітки: К, Г, Т.

66. Частина 2. Реалізація модуля для прораховування траєкторії стріли залежно від кута, сили запуску, напрямку та сили вітру.

Програма повинна реалізовувати прораховування траєкторії стріли і координати в будь-який момент часу, якщо відомо силу та кут нахилу лука. Швидкість вітру може бути як детермінована, так і випадкова величина, яка завжди відома і змінює траєкторію стріли у польоті. Програма повинна передавати дані модулям із частини 1 та частини 3. Обов'язкове використання розрахунку фізики з модуля Unity.

Вхідні дані: сила та кут натягу лука, швидкість і напрям вітру.

Вихідні дані: поточні координати і траєкторія стріли.

Складність: середня.

Примітки: К, М, Т.

67. Частина 3. Реалізація модуля для створення перешкод (непробивні блоки, відбивачі, важелі, механізми).

Програма повинна реалізовувати перешкоди для стріли(непробивні блоки, відбивачі, важелі, механізми), задавати властивості та розмір перешкод кожного типу, та прораховування вплив на траєкторію стріли даних перешкод(поточні координати і траєкторія отримуються від модуля частини 2). Обов'язкове використання розрахунку для фізики з модуля Unity.

Вхідні дані: ім'я користувача, кількість пройдених рівнів, сила та кут натягу лука, швидкість і напрям вітру, кількість, параметри і властивості перешкод.

Вихідні дані: поточні координати і траєкторія стріли з урахуванням перешкод.

Складність: середня.

Примітки: К, М, Т.

68. Частина 4. Реалізація набору рівнів гри та етапу навчання.

Програма повинна реалізовувати створення рівнів гри із наростанням складності, в яких відбувається розстановка перешкод. Також потрібно передбачити навчальну кампанію для гравця, який не має елементарних знань з фізики та уявлення про техніку стрільби із лука із поясненнями.

Вхідні дані: кількість рівнів, параметри і властивості перешкод.

Вихідні дані: файли рівнів із розставленими перешкодами, файл опису(допомоги).

Складність: середня.

Примітки: К, Г.

Реалізація гри «мушкетер».

69. Частина 1. Реалізація інтерфейсу користувача(GUI).

Програма повинна реалізовувати графічний інтерфейс користувача(GUI), який реалізує графічно власне ігровий процес: фігуру гравця-стрільця, який може змінювати кількість пороху, тип зброї та кут нахилу; куля у польоті(розрахунок координат, траєкторії, вплив вітру виконує розробник частини 2); малювання різних перешкод (властивості перешкод та їх вплив на політ кулі залежно від кута влучання виконує розробник частини 3), а також можливість рикошету, роздроблення кулі, появу осколків перешкод; надає можливість введення користувача, завантаження його даних і статистики; відображає положення і дію ворогів(розрахунок їх позиції, поведінки, стратегії виконує розробник частини 4). Завдання розробника частини 1 полягає у створенні графічної бібліотеки, яку повинні використовувати розробники інших модулів. Обов'язкове використання хоча б однієї власної текстури/спрайта.

Вхідні дані: ім'я користувача, кількість пройдених рівнів, тип зброї, кількість і тип ворогів, перелік перешкод і їх типу(матеріалу), швидкість і напрям вітру.

Складність: висока.

Примітки: К, Г.

70. Частина 2. Реалізація модуля для прораховування траєкторії кулі залежно від кута, кількості пороху, напрямку та сили вітру.

Програма повинна реалізовувати прораховування траєкторії кулі і координати в будь-який момент часу, якщо відомо кількість зарядженого пороху та кут нахилу зброї, а також випадковий параметр дрижання зброї для недосвідченого гравця. Швидкість вітру може бути як детермінована, так і випадкова величина, яка завжди відома і змінює траєкторію стріли у польоті. Програма повинна передавати дані модулям із частини 1, 3, 4. Обов'язкове використання одного з існуючих ігрових рушіїв(або його елементів), наприклад Unreal Engine.

Вхідні дані: тип зброї, кількість заряду пороху, кут нахилу зброї, напрям та величина коливання зброї, швидкість і напрям вітру.

Вихідні дані: поточні координати і траєкторія кулі.

Складність: середня.

Примітки: К, М, Т.

71. Частина 3. Реалізація модуля властивостей матеріалів поверхонь(міцність, крихкість, можливість рикошету та роздроблення) та розрахунок імпульсу кулі до і після влучання.

Програма повинна реалізовувати перешкоди для кулі(непробивні блоки, відбивачі, пробивні блоки, блоки, що розсипаються), задавати властивості та розмір перешкод кожного типу, та прораховування вплив на траєкторію кулі даних перешкод(поточні координати і траєкторія отримуються від модуля частини 2). Також повинна бути розрахована імовірність рикошету та роздроблення кулі, швидкість, напрям та імпульс уламків, ступінь знищення та імовірність пробиття перешкоди. Обов'язкове використання розрахунку фізики з модуля Unity.

Вхідні дані: тип зброї, кількість заряду пороху, кут нахилу зброї, імпульс та швидкість кулі, тип(матеріал) та розмір перешкоди, ступінь знищення перешкоди.

Вихідні дані: поточні координати і траєкторія кулі з урахуванням перешкод, імовірність рикошету та роздроблення кулі, швидкість, напрям та імпульс уламків, ступінь знищення та імовірність пробиття перешкоди.

Складність: середня.

Примітки: К, М, Т.

72. Частина 4. Реалізація модуля штучного інтелекту для ворогів (різних типів).

Програма повинна реалізовувати створення штучного інтелекту, який задає тактику і стратегію для ворогів різних типів. Штучний інтелект повинен вибирати поведінку залежно від типу зброї гравця і власної зброї, наявності, розміру і типу перешкод. У кожного ворога є свій тип(стратегія) поведінки. Бажано додати можливість самонавчання «сильних» ворогів. . Обов'язкове використання одного з існуючих ігрових рушіїв (наприклад з модуля Unity).

Вхідні дані: поточний рівень, статистика успішності гравця і ворогів, наявність та тип перешкод, тип зброї гравця і ворогів.

Вихідні дані: координати положення ворогів(куди йти), час пострілів, кут нахилу зброї для кожного з ворогів.

Складність: висока.

Примітки: К, М.

73. Частина 5. Реалізація модуля вдосконалень для зброї гравця та ворогів.

Програма повинна реалізовувати створення можливості вдосконалення зброї гравця, ворогів залежно або незалежно від попередніх успіхів. Повинен враховуватись баланс сили гравця та ворогів. Розробник задає параметри зброї, які повинні використовувати для своїх розрахунків розробники всіх інших частин(1-5).

Вхідні дані: кількість рівнів, статистика успішності гравця і ворогів.

Вихідні дані: параметри вдосконаленої зброї(імовірність дрижання, кількість пороху, початкова швидкість кулі, час перезарядки).

Складність: середня.

Примітки: К.

74. Реалізація програми з віртуальною реальністю(VR).

Програма повинна реалізовувати на створення віртуальною реальності на камері комп'ютера, мобільного телефона, планшета. Зображення зробити

самостійно в 3D чи VR-редакторі. Зображення повинно змінюватись залежно від положення та повороту камери.

Складність: середня.

Примітки: Г.

75. Реалізація програми з доповненою реальність (AR) на мобільному пристрої.

Програма повинна реалізовувати на мобільному телефоні або планшеті програму для створення доповненої реальності. Зображення, яке виникає при наведенні на опорну точку, зробити самостійно в 3D-редакторі. Зображення повинно бути рухомим. Дозволяється використовувати будь-які засоби(в т.ч. набори AR Studio), окрім модуля Unity 3D. Корисною буде робота з бібліотекою Vuforia.

Складність: середня.

Примітки: Г.

76. Реалізація програми з доповненою реальність(AR) з допомогою Unity 3D

Програма повинна реалізовувати на створення доповненої реальності при наведенні на опорну точку камерою комп'ютера, мобільного телефона, планшета. Зображення зробити самостійно в 3D-редакторі. Зображення повинно бути інтерактивним(взаємодіяти з користувачем через пристрій введення чи екранну клавіатури). Обов'язкове використання Unity 3D. Корисною буде робота з бібліотекою Vuforia та фізикою в Unity.

Складність: середня.

Примітки: Г, Т.

77. Реалізація шпигунської програми, замасковану під гру з доповненою реальність(AR).

Програма повинна реалізовувати гру, принцип якої аналогічний Pokemon Go, тільки в менших масштабах: збираєш деякі елементи, створені на основі доповненої реальності і отримуєш за це віртуальну винагороду. Елементи генеруються випадково чи за заданим алгоритмом. В процесі гри має бути передбачена можливість таємного передавання фото/відео з камери за вказаною адресою. У адміністратора програми повинна бути можливість «зйомки» потрібних місць карти (через камери користувачів з цією програмою, які грають у вказаних межах), редагування положення і кількості ігрових елементів, можливість надсилання повідомлення користувачам через гру. Корисною буде робота з бібліотекою Vuforia та фізикою в Unity.

Складність: висока.

Примітки: Г, З.

78. Реалізація гри «симулятор каменю»

Програма повинна реалізувати вигляд для спостерігача відносно деякої нерухомої точки та відобразити на екрані зображення природних чи штучних об'єктів з камери/спостерігача (якість не має значення, можна використовувати підхід низько полігональних моделей LowPoly). Наприклад, вид відносно «каменю». Усі необхідні зображення, текстури, спрайти, моделі зробити самостійно в 3D-редакторі. У грі повинні бути як рухомі моделі, так і нерухомі (положення

спостерігача(каменя) незмінне). Передбачити можливість зміни швидкості часу та запис «історії життя каменя.

Складність: середня.

Примітки: Г.

79. Реалізація гри «лабіринт»

Програма повинна гру, де головний герой ходить по лабіринту у шукає вихід. У лабіринті кожна стіна має свою текстуру/спрайт(якщо зустрічаються 2 протилежні стіни однакової довжини, на доній зображення повинно бути створене векторним редактором, на протилежній-расторовим). Усі необхідні зображення, текстури, спрайти, моделі зробити самостійно. Гра повинна мати вигляд 3D (дозволяється використовувати підхід «псевдо 3D»). Вихід з лабіринту для головного героя повинен бути анімованим.

Складність: низька.

Примітки: Г.

80. Реалізація гри на основі ігрового рушія Minecraft

Програма повинна гру, де головний герой ходить по території, яка може редагуватись. Рельєф може складатися з окремих моделей та мати різні текстури/спрайти. Гра повинна мати вигляд 3D (дозволяється використовувати підхід «псевдо 3D»). Для гри можна використовувати підхід низько полігональних моделей LowPoly та ігровий рушія Minecraft або аналогічний, наприклад з Unity.

Складність: низька.

Примітки: Г.

Додаток А
Зразки індивідуальних завдань до курсових робіт

Індивідуальне завдання № _____

На курсову роботу з курсу “Програмування”

Розробити: _____

Вимоги до розробки:

1. _____
2. _____
3. _____
4. _____

Примітка:

1. _____.
2. _____.
3. _____.
4. _____.

Затверджено на засіданні кафедри _ вересня 201_ р. Протокол № _

Дата видачі _____ вересня 201_ р.

Керівник роботи

Завдання отримав