

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ ДЕРЖАВНИЙ ПОЛІТЕХНІЧНИЙ
КОЛЕДЖ

для спеціалізації
5.123.1 «Обслуговування
комп'ютерних систем і мереж»

Методичний посібник
для курсового проектування
з навчальної дисципліни

Програмування

2019 р.

Методичний посібник для курсового проектування з навчальної дисципліни “Програмування” для студентів спеціальності 123 "Комп'ютерна інженерія", спеціалізації 5.123.1 "Обслуговування комп'ютерних систем і мереж".

Розробник: Ярмола О. С., викладач 1 категорії.

Методичний посібник для курсового проектування затверджений на засіданні циклової комісії інформаційних технологій.

Протокол № ____ від «__» _____ 20__ року

Голова циклової комісії інформаційних технологій

_____ М.М. Бочарніков

(підпис)

Схвалено методичною радою Харківського державного політехнічного коледжу

Протокол № ____ від «__» _____ 20__ року

Голова методичної ради коледжу _____ В.О. Величко

(підпис)

ЗМІСТ

ВСТУП	4
1 ПОСЛІДОВНІСТЬ ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ	5
1.1 Вимоги до змісту пояснювальної записки	6
1.2 Стадії розробки програмного забезпечення	7
1.3 Оформлення роботи	9
1.4 Захист курсового проекту	9
2 РЕКОМЕНДАЦІЇ ЩОДО ОФОРМЛЕННЯ РОБІТ	10
2.1 Загальні вимоги	10
2.2 Нумерація	11
2.3 Ілюстрації	12
2.4 Таблиці	13
2.5 Формули	14
2.6 Загальні правила цитування та посилання на використані джерела	15
2.7 Оформлення списку використаних джерел	16
2.8 Додатки	16
3 РЕКОМЕНДАЦІЇ ЩОДО ЗМІСТУ ПРОЕКТУ	17
4 ОРІЄНТОВНЕ ЗАВДАННЯ НА КУРСОВЕ ПРОЕКТУВАННЯ	25
ЛІТЕРАТУРА	34
ДОДАТКИ	35

ВСТУП

На цей час розвитку суспільства інформаційні технології є невід'ємною частиною життя та професійної діяльності людини. Вже не викликає сумніву теза про необхідність глибокого знання основ архітектури комп'ютера, комп'ютерної мережі та програмного забезпечення. Тому вивченню теоретичного фундаменту та наданню практичних навичок роботи з сучасними офісними та професійними програмними продуктами в навчальних планах підготовки спеціалістів різних фахів надається все більше уваги.

Одним з основних видів навчальної роботи з дисциплін інформаційної спрямованості є виконання курсового проекту, що дає змогу застосувати одержані теоретичні знання для розв'язання конкретної задачі.

Дисципліни інформаційної спрямованості викладаються протягом усього циклу навчання, починаючи з першого курсу, тому для якісної підготовки майбутнього інженера необхідним є вміння послідовно та логічно викласти матеріал, правильно сформулювати технічне завдання на розробку чи застосування програмного продукту, оформити курсовий проект згідно нормативних документів.

Дані методичні вказівки призначені забезпечити процес виконання та захисту курсової роботи та надати знання та вміння складання та оформлення звітної документації, які стануть у нагоді під час написання студентом дипломного проекту та подальшій професійній діяльності, містить єдині вимоги щодо виконання курсових проектів для студентів денної форми навчання.

1 ПОСЛІДОВНІСТЬ ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ

Курсовий проект з дисципліни «Програмування» виконується студентами в такій послідовності:

- вивчення спеціальної літератури за темою роботи;
- складання плану;
- написання програми на комп'ютері;
- виконання розрахунків та аналіз результатів;
- написання та оформлення роботи;
- підготовка до захисту та захист курсової роботи.

При виборі теми проекту враховуються схильність та інтерес студента до вивчення певної проблеми, що потребує самостійного вирішення даного питання. Допускається виконання одного завдання кількома студентами. Теми робіт, обрані студентами, та склад підгруп узгоджуються з керівником. Студент може запропонувати власну тему курсового проекту, але вона повинна бути узгоджена з керівником.

Після одержання завдання студент повинен зробити аналітичний огляд сучасної наукової літератури за темою проекту, зокрема в періодичних виданнях. Потім складається план, який узгоджується з керівником.

Програма, що створюється, має бути написана за допомогою програмного забезпечення, яке встановлено (або може бути встановленим) в лабораторії.

Для виконання курсового проекту необхідно розробити пакет документів і реалізувати на комп'ютері поставлену задачу. Пакет документів повинен містити:

- пояснювальну записку;
- технічне завдання.

1.1 Вимоги до змісту пояснювальної записки

1.1.1 **Загальні положення.** Пояснювальна записка до курсового проекту повинна складатися з:

- титульного листа (1 сторінка), який видається керівником курсового проектування;
- листа завдання на курсове проектування (2 сторінки);
- змісту пояснювальної записки (1-2 сторінки);
- переліку умовних скорочень, якщо в цьому є необхідність (1 сторінка);
- вступу (1-2 сторінки);
- постановки та аналізу завдання (1 сторінка);
- теоретичної частини (5-10 сторінок);
- опису програмного забезпечення (5-10 сторінок);
- питань охорони праці та техніки безпеки (3-10 сторінок);
- висновків (1 сторінка);
- списку використаних джерел (1 сторінка);
- додатків.

Загальний обсяг пояснювальної записки повинен бути не менше 15-20 машинописних сторінок або 20-25 сторінок рукописного тексту.

Окремі розділи (підрозділи) документа припускається об'єднувати, а також вводити нові розділи (підрозділи).

1.1.2 **Зміст розділів.** У вступі висвітлюються літературні джерела, які були використані при написанні даного проекту, наводяться вступні положення теорії, обґрунтовується актуальність вибраної теми.

Лист завдання, яку підписано викладачем із затвердженою датою видачі завдання на курсове проектування, повинна містити чітке, однозначне та структуроване формулювання задачі на курсове проектування.

Теоретична частина може містити в собі кілька розділів. У теоретичній частині висвітлюються всі теоретичні питання, необхідні для розв'язання завдання, яке сформульовано в розділі ПОСТАНОВКА ЗАВДАННЯ, наводяться формули, схеми.

У розділі ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ потрібно привести блок-схеми програми або словесний опис функціонування програми (текст програми виноситься в додатки), призначення й область застосування програмного забезпечення, що розробляється, основні технічні характеристики.

Типовий вміст розділу ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ є таким:

1. Підрозділ "Призначення й область застосування", у якому потрібно описати призначення програми і стисло характеристику можливої області її застосування.
2. Підрозділ "Технічні характеристики" з такими пунктами:
 - постановка задачі на розробку програми, опис застосованих математичних методів і, при необхідності, допущень і обмежень, пов'язаних з обраним математичним апаратом або програмною системою обробки інформації.
 - опис алгоритму і (або) функціонування програми з обґрунтуванням вибору схеми алгоритму рішення задачі, можливі взаємодії програми з іншими програмами.
 - опис і обґрунтування вибору:
 - а) методу організації вхідних і вихідних даних;
 - б) складу технічних і програмних засобів.
 - розподіл носіїв даних, що використовує програма, на підставі проведених розрахунків і (або) аналізів.

ВИСНОВКИ повинні містити в собі аналіз результатів, які отримані у попередньому розділі.

У розділі СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ необхідно зазначити перелік науково-технічних публікацій, нормативно-технічних документів і інших матеріалів, на які є посилання в основному тексті.

Список використаних джерел оформляється у відповідності з прийнятими стандартами.

У додатки до пояснювальної записки повинен бути винесений лістинг програмного коду.

1.2 Стадії розробки програмного забезпечення

Стадії розробки програмного забезпечення наведено у табл.1.1.

Таблиця 1.1 Стадії розробки програмного забезпечення

Стадії розробки	Етапи робіт	Зміст робіт
Технічне завдання (ТЗ)	Обґрунтування необхідності розробки програми	Постановка задачі. Збір вихідних матеріалів. Вибір й обґрунтування критеріїв ефективності і якості програми, що розробляється.
	Науково-дослідні роботи	Визначення структури вхідних і вихідних даних. Попередній вибір методів розв'язання задач. Обґрунтування доцільності використання програм, що розроблені раніше. Визначення вимог до технічних засобів. Обґрунтування принципової можливості розв'язання поставленої задачі.
	Розробка та затвердження технічного завдання	Визначення вимог до програми. Визначення стадій, етапів, термінів розробки програми і документації до неї. Вибір мов програмування. Узгодження та затвердження технічного завдання.
Ескізний проект	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження та затвердження ескізного проекту.
	Розробка програми	Програмування й налагодження програми.
Робочий проект	Розробка програмної документації	Розробка програмних чи графічних документів відповідно до вимог стандартів.
	Випробування програми	Коректування програми і програмної документації відповідно до результатів випробувань.
Упровадження		

1.3 Оформлення роботи

Робота має бути оформлена на сторінках формату А4 у відповідності зі стандартами щодо оформлення курсових проектів.

Обов'язковою умовою є наявність лістингу всіх модулів проекту (в додатку) та скрінів розробленого додатку під час виконання кожного із модулів (в пояснювальній записці). Електронна версія пояснювальної записки, вихідні файли, файл, що виконується, програма, яку розроблено, надається викладачеві.

1.4 Захист курсового проекту

Захист виконаного, відповідним чином оформленого і підписаного керівником та узгодженого зав. відділенням та зав. циклової комісії курсового проекту проводиться перед комісією з трьох викладачів у встановлений строк. Зміст, оформлення та захист проекту оцінюється комісією з виставленням оцінки.

2 РЕКОМЕНДАЦІЇ ЩОДО ОФОРМЛЕННЯ РОБІТ

2.1 Загальні вимоги

Роботу пишуть вручну або друкують за допомогою комп'ютера на одній стороні аркуша білого паперу формату А4 (210x297мм). Мінімальна висота шрифту 1,8 мм.

Параметри сторінки в текстовому редакторі Word:

- верх - 15 мм;
- низ - 30 мм;
- зліва - 25 мм;
- справа - 10 мм.

Параметри абзацу в текстовому редакторі Word:

- відступ - 12,7 мм;
- міжрядковий інтервал полуторний;
- вирівнювання по ширині;
- відступи й інтервали - 0 мм;
- заборона висячих рядків.

Параметри шрифту в текстовому редакторі Word:

- тип - Times New Roman;
- розмір - 14;
- креслення - звичайний.

Параметри списку аналогічні параметрам абзацу, нумерація без крапки, при перерахуванні дозволяється маркірувати абзац рисою.

Шрифт друку повинен бути чітким, строчка - чорного кольору середньої жирності. Щільність тексту роботи повинна бути однаковою.

Роздруковані на комп'ютері програмні документи або графічні ілюстрації повинні відповідати формату А4, їх включають до загальної нумерації сторінок роботи і розміщують, як правило, в додатках.

Текст основної частини роботи поділяють на розділи, підрозділи, пункти та підпункти.

Заголовки структурних частин роботи «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ», «ВСТУП», «ПОСТАНОВКА ЗАВДАННЯ», «РОЗДІЛ», «ВИСНОВКИ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», «ДОДАТКИ» друкують великими літерами симетрично до тексту. Заголовки підрозділів друкують маленькими літерами (крім першої великої) з абзацного відступу. Крапку в кінці заголовка не ставлять. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою. Заголовки підпунктів друкують маленькими літерами (крім першої великої) з абзацного відступу. В кінці заголовка не ставиться крапка.

Відстань між заголовком (за винятком заголовка пункту) та текстом повинна дорівнювати 1-2 інтервалам.

2.2 Нумерація

Нумерацію сторінок, розділів, підрозділів, пунктів, підпунктів, малюнків, таблиць, формул подають арабськими цифрами без знака №.

Першою сторінкою роботи є титульний аркуш, другий – аркуш завдання на курсове проектування, які включають до загальної нумерації сторінок роботи. На вказаних аркушах номер сторінки не ставлять, на наступних сторінках номер проставляють у відведеному місці штампу без крапки в кінці. На аркуші 3 міститься зміст роботи, який бажано створювати в автоматичному режимі за допомогою засобів текстового редактора.

Такі структурні частини роботи, як зміст, перелік умовних позначень, вступ, постановка завдання, висновки, список використаних джерел не мають порядкового номера. Всі аркуші, на яких розміщені згадані структурні частини роботи, нумерують звичайним чином. Не нумерують лише їх заголовки, тобто не можна друкувати: «1 ВСТУП» або «6 ВИСНОВКИ». Номер розділу ставлять перед назвою розділу, після номера крапку не ставлять.

Підрозділи нумерують у межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, між якими ставлять крапку. В кінці номера підрозділу не повинна стояти крапка, наприклад: «2.3» (третій підрозділ другого розділу). Потім у тому ж рядку йде заголовок підрозділу.

Пункти нумерують у межах кожного підрозділу. Номер пункту складається з порядкових номерів розділу, підрозділу, пункту, між якими ставлять крапку. В кінці номера не повинна стояти крапка, наприклад: «1.3.2» (другий пункт третього підрозділу першого розділу). Потім у тому ж рядку йде заголовок пункту. Пункт може не мати заголовка.

Підпункти нумерують у межах кожного пункту за такими ж правилами, як пункти.

Ілюстрації (фотографії, креслення, схеми, графіки, карти) і таблиці необхідно подавати в роботі безпосередньо після тексту, де вони згадані вперше, або на наступній сторінці. Ілюстрації і таблиці, які розміщені на окремих сторінках роботи, включають до загальної нумерації сторінок. Таблицю, рисунок або креслення, розміри якого більше формату А4, враховують як одну сторінку і розміщують у відповідних місцях після згадування в тексті або у додатках.

Ілюстрації позначають словом «Рис.» і нумерують послідовно в межах розділу, за винятком ілюстрацій, поданих у додатках.

Номер ілюстрації повинен складатися з номера розділу і порядкового номера ілюстрації, між якими ставиться крапка.

Наприклад:

Рис. 1.2 (другий рисунок першого розділу).

Номер ілюстрації, її назва і пояснювальні підписи розміщуються послідовно під ілюстрацією. Якщо в роботі подано одну ілюстрацію, то її нумерують за загальними правилами.

Таблиці нумерують послідовно (за винятком таблиць, поданих у додатках) в межах розділу. В правому верхньому куті над відповідним заголовком таблиці розміщують напис «Таблиця» із зазначенням її номера. Номер таблиці повинен складатися з номера розділу і порядкового номера таблиці, між якими ставиться крапка, наприклад: «Таблиця 1.2» (друга таблиця першого розділу).

Якщо в роботі одна таблиця, її нумерують за загальними правилами.

При переносі частини таблиці на інший аркуш (сторінку) слово «Таблиця» і номер її вказують один раз справа над першою частиною таблиці, над іншими частинами пишуть слова «Продовження табл.» і вказують номер таблиці, наприклад: «Продовження табл. 1.2».

Формули в роботі (якщо їх більше одної) нумерують у межах розділу. Номер формули складається з номера розділу і порядкового номера формули в розділі, між якими ставлять крапку. Номери формул пишуть біля правого берега аркуша на рівні відповідної формули в круглих дужках, наприклад: (3.1) (перша формула третього розділу).

Примітки до тексту і таблиць, в яких вказують довідкові і пояснювальні дані, нумерують послідовно в межах однієї сторінки. Якщо приміток на одному аркуші декілька, то після слова «Примітки» ставлять двокрапку, наприклад:

Примітки:

1. ...
2. ...

Якщо є одна примітка, то її не нумерують і після слова. «Примітка» ставлять крапку.

Текст пояснювальної записки оформляється згідно вимог. Приклад оформлення дивись в додатках.

2.3 Ілюстрації

Ілюструють роботу, виходячи із певного загального задуму, за ретельно продуманим тематичним планом, який допомагає уникнути ілюстрацій

випадкових, пов'язаних із другорядними деталями тексту і запобігти невинуватим пропускам ілюстрацій до найважливіших тем. Кожна ілюстрація має відповідати тексту, а текст - ілюстрації.

Назви ілюстрацій розміщують після їхніх номерів. При необхідності ілюстрації доповнюють пояснювальними даними (підрисунковий підпис).

Підпис під ілюстрацією звичайно має чотири основних елементи:

- найменування графічного сюжету, що позначається скороченим словом «Рис. »;
- порядковий номер ілюстрації, який вказується без знаку номера арабськими цифрами;
- тематичний заголовок ілюстрації, що містить текст із якомога стислою характеристикою зображеного;
- експлікацію, яка будується так: деталі сюжету позначають цифрами, які виносять у підпис, супроводжуючи їх текстом. Потрібно зазначити, що експлікація не замінює загального найменування сюжету, а лише пояснює його.

Приклад:

Рис. 1.24. Схема розміщення елементів:

- 1 – елемент 1;
- 2 – елемент 2;
- 3 – елемент 3.

Основними видами ілюстративного матеріалу в роботі є: креслення, технічний рисунок, схема, фотографія, діаграма і графік.

Не варто оформлювати посилання на ілюстрації як самостійні фрази, в яких лише повторюється те, що міститься у підписі. У тому місці, де викладається тема, пов'язана із ілюстрацією, і де читачеві потрібно вказати на неї, розміщують посилання у вигляді виразу у круглих дужках «(рис. 3.1) » або зворот типу: «...як це видно з рис. 3.1» або «... як це показано на рис. 3.1».

Якість ілюстрацій повинна забезпечувати їх чітке відтворення.

2.4 Таблиці

Цифровий матеріал, як правило, повинен оформлятися у вигляді таблиць.

Приклад побудови таблиці

		Таблиця (номер)		Назва таблиці	
Заголовок		Заголовки		Заголовки	
		Підзаголовки	Підзаголовки	Підзаголовки	Підзаголовки
Рядки	Заголовки граф				
Боковик (заголовки рядків)		Графи (колонки)			

Кожна таблиця повинна мати назву, яку розміщують над таблицею і друкують симетрично до тексту. Назву і слово «Таблиця» починають з великої літери. Назву не підкреслюють.

Заголовки граф повинні починатися з великих літер, підзаголовки - з маленьких, якщо вони складають одне речення із заголовком, і з великих, якщо вони є самостійними. Висота рядків повинна бути не меншою 8 мм. Графу з порядковими номерами рядків до таблиці включати не обов'язково.

На всі таблиці роботи повинні бути посилання в тексті, при цьому слово «таблиця» в тексті пишуть скорочено, наприклад: «...в табл. 1.2».

У повторних посиланнях на таблиці та ілюстрації потрібно вказувати скорочено слово «дивись», наприклад: «див.табл. 1.3».

Таблицю розміщують після першого згадування про неї в тексті, таким чином, щоб її можна було читати без повороту переплетеного блоку роботи або з поворотом за годинниковою стрілкою. Таблицю з великою кількістю рядків можна переносити на інший аркуш. При перенесенні таблиці на інший аркуш (сторінку) назву вміщують тільки над її першою частиною. Таблицю з великою кількістю граф можна ділити на частини і розміщувати одну частину під іншою в межах одної сторінки.

Якщо цифрові або інші дані в якому-небудь рядку таблиці не подають, то в ньому ставлять прочерк.

2.5 Формули

При використанні формул необхідно дотримуватися певних техніко-орфографічних правил. Найбільші, а також довгі і громіздкі формули, які мають у складі знаки, суми, добутку, диференціювання, інтегрування, розміщують на окремих рядках. Це стосується також і всіх нумерованих формул. Для економії місця кілька коротких однотипних формул, відокремлених від тексту, можна подати в одному рядку, а не одну під одною. Невеликі і нескладні формули, що не мають самостійного значення, вписують всередині рядків тексту.

Пояснення значень символів і числових коефіцієнтів необхідно подавати безпосередньо під формулою в тій послідовності, в якій вони дані у формулі. Значення кожного символу і числового коефіцієнта потрібно подавати з нового рядка. Перший рядок пояснення починають зі слова «де» без двокрапки.

Рівняння і формули потрібно виділяти з тексту вільними рядками. Вище і нижче кожної формули необхідно залишити не менше одного вільного рядка. Якщо рівняння не вміщується в один рядок, його слід перенести після знака рівності (=) або після знаків плюс (+), мінус (-), множення (x) і ділення (:).

Нумерувати слід лише ті формули, на які є посилання у наступному тексті. Інші нумерувати не рекомендується.

Порядкові номери позначають арабськими цифрами в круглих дужках біля правого краю сторінки без крапок від формули до її номера. Номер, який не вміщується у рядку з формулою, переносять у наступний нижче формули. Номер формули при її перенесенні вміщують на рівні останнього рядка.

Посилання на формули роботи вказують порядковим номером формули в дужках, наприклад "... у формулі (2.1)".

2.6 Загальні правила цитування та посилання на використані джерела

При написанні роботи студент повинен давати посилання на джерела, матеріали або окремі результати, які наводяться в роботі, або на ідеї і висновки, на основі яких розроблюються проблеми, задачі, питання, вивченню яких присвячена робота. Такі посилання дають змогу відшукати документи і перевірити достовірність відомостей про цитування документа, дають необхідну інформацію щодо нього, допомагають з'ясувати його зміст, мову тексту, обсяг. Посилатися слід на останні видання публікацій. На більш ранні видання можна посилатися лише в тих випадках, коли в них наявний матеріал, який не включено до останнього видання.

Якщо використовують відомості, матеріали з монографій, оглядових статей, інших джерел з великою кількістю сторінок, тоді в посиланні

необхідно точно вказати номери сторінок, ілюстрацій, таблиць, формул з джерела, на яке дано посилання в роботі.

Посилання в тексті роботи на джерела слід зазначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад, «... у працях [1-7]...».

Якщо в тексті роботи необхідно зробити посилання на складову частину або на конкретні сторінки відповідного джерела, можна наводити посилання у виносках, при цьому номер посилання має відповідати його бібліографічному опису за переліком посилань.

2.7 Оформлення списку використаних джерел

Список використаних джерел - елемент бібліографічного апарату, який містить бібліографічні описи використаних джерел і розміщується після висновків.

Бібліографічний опис складають безпосередньо за друкованим текстом або виписують з каталогів і бібліографічних покажчиків повністю без пропусків будь-яких елементів, скорочення назв і т. ін. Завдяки цьому можна уникнути повторних перевірок, вставок пропущених відомостей.

Джерела можна розміщувати одним із таких способів: у порядку появи посилань у тексті (найбільш зручний для користування і рекомендований при написанні роботи), в алфавітному порядку прізвищ перших авторів або заголовків, у хронологічному порядку.

2.8 Додатки

Додатки оформляють як продовження роботи на наступних її сторінках, розміщуючи їх у порядку появи посилань у тексті роботи.

Кожний такий додаток повинен починатися з нової сторінки. Додаток повинен мати заголовок, надрукований угорі малими літерами з першої великої симетрично відносно тексту сторінки. Посередині рядка над заголовком великими літерами друкується слово «ДОДАТОК» і номер додатку.

З РЕКОМЕНДАЦІЇ ЩОДО ЗМІСТУ ПРОЕКТУ

Під час виконання курсового проекту студент розробляє програмний код та оформлює пояснювальну записку. Програмний код створюється або в середовищі програмування Delphi (версія не нижче Delphi7) або за допомогою мови програмування Pascal. Приблизний вміст пояснювальної записки:

ВСТУП

1 ПОСТАНОВКА ЗАВДАННЯ

1.1 АНАЛІЗ ЗАВДАННЯ, ВИБІР МЕТОДІВ ОБРОБКИ ДАНИХ

1.2 СТВОРЕННЯ АЛГОРИТМУ РОБОТИ ПРОГРАМИ

2 СКЛАД СЕРЕДОВИЩА ПРОГРАМУВАННЯ

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 ОПИС МЕТОДІВ, КОТРІ БУЛИ ВИКОРИСТАНІ ПРИ РОЗРОБЦІ ПРОГРАМИ

3.2 РОБОТА З ПРОЕКТОМ

4 ОХОРОНА ПРАЦІ І ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ

ВИСНОВОК

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ (ГРАФІЧНА ЧАСТИНА)

Пункт 1 – постановка завдання, повинен містити в собі розшифровку завдання, а також проаналізовано, які структури даних, компоненти середовища, процедури і функції ми будемо використовувати під час роботи.

Наприклад:

1 ПОСТАНОВКА ЗАВДАННЯ

Варіант № ____.

Завдання:

Вивести в порядку зростання всі нескоротні дроби, укладені між 0 і 1 знаменники яких не перевищує N. Обмеження: $2 < N < 255$.

Організувати можливість введення і виведення інформації у файл і за допомогою екранної форми.

Під час виконання курсового проекту я повинен розробити алгоритм роботи програми, створити інтерфейс в середовищі Delphi, написати програмний код для виконання завдання.

1.1 АНАЛІЗ ЗАВДАННЯ, ВИБІР МЕТОДІВ ОБРОБКИ ДАНИХ

Робота програми виконується згідно наступної схеми:

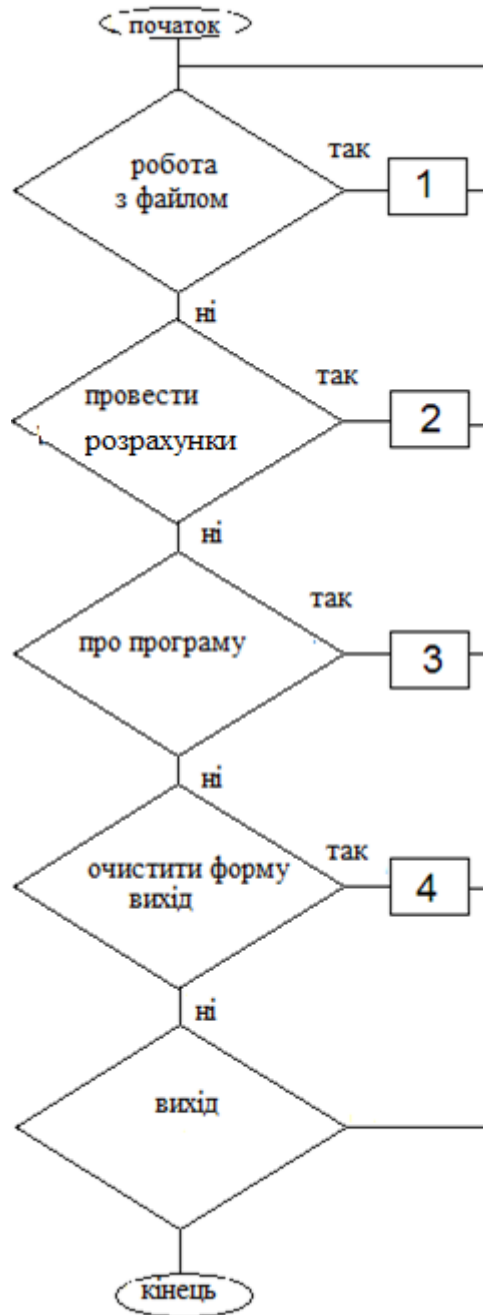


Рис. 1.1 Схема роботи програми

На екран виводиться форма, на якій розташовані: меню та кнопка "Розрахунок". В меню містяться пункти "Файл", "Про програму" і "Вихід".

Склад головного меню додатка зображено на рисунку.



Рис. 1.2 Склад головного меню додатка

При натисканні на меню, на пункт "Файл", користувачеві стають доступні такі підпункти пункти меню, як "Відкрити" і "Зберегти". Якщо буде обраний перший пункт, то дані для розрахунку, які відіб'ються на екрані, будуть взяті з файлу, який вибере користувач. Якщо буде обраний другий пункт, то дані з екрану будуть збережені у файл, який вибере користувач.

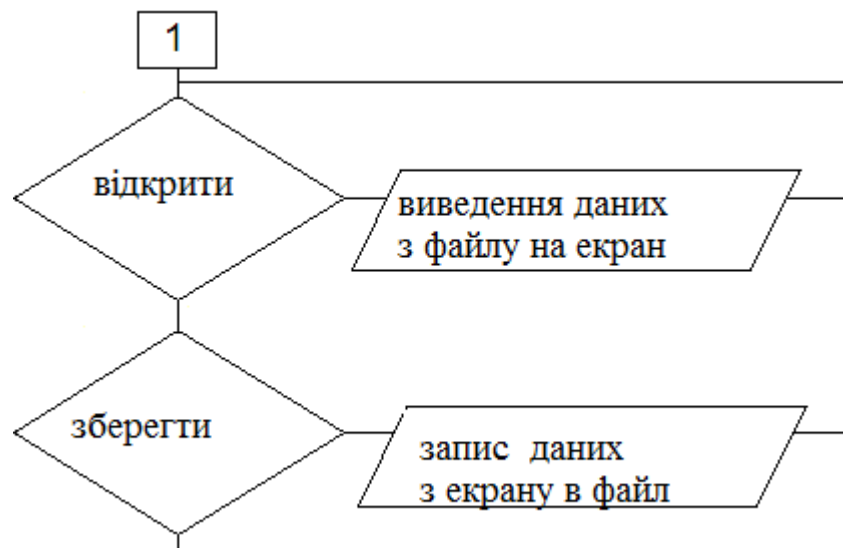


Рис. 1.3 Схема роботи пункту меню «Файл»

Таким чином студент повинен описати структуру програми, вказати методи, які він застосовував. Кожен пункт меню, кожна процедура повинна бути описана. Схема роботи програми може бути подана в довільному вигляді (схематичний, графічний чи словесний спосіб за вибором студента.)

Після опису принципу роботи програми студент повинен створити блок-схему роботи програми, а також навести блок-схеми роботи кожної процедури та функції. Крім того, дозволяється давати додатково словесний опис алгоритму. Кожна блок-схема повинна бути підписана.

Алгоритм має задовольняти певним вимогам, серед яких потрібно виділити найважливіші.

Визначеність — кожен крок алгоритму має інтерпретуватися виконавцем однозначно.

Результативність — за скінченну кількість кроків алгоритм має приводити до розв'язання задачі або зупинятися через неможливість її розв'язати.

Дискретність — кроки обчислювального процесу мають бути відокремлені один від одного.

Ефективність — під час розв'язання задачі може використовуватися лише обмежений обсяг комп'ютерних ресурсів.

Масовість — алгоритм розробляється у загальному вигляді, тобто його можна застосувати не лише до окремої задачі, але і до деякого класу задач, що розрізняються лише вхідними даними.

Всі символи блок-схеми алгоритму повинні сполучатися лініями зі стрілками, які вказують на зв'язок між блоками та визначають, до якого блоку здійснюється перехід після виконання дій даного блоку. У випадку, коли напрямок дій зверху вниз, стрілки на кінцях ліній можна не ставити.

Зв'язки між символами блок-схеми алгоритму, розміщеними на різних місцях одного або різних аркушів можуть зображатися не тільки лініями, а і спеціальним символом-сполучником, всередині якого вказується або номер символу, до входу якого підключається вихід даного символу, або номер символу, вихід якого підключається до входу даного символу.

Після опису схеми роботи і надання блок-схеми студенти повинні розробити програму. В пояснювальній записці до курсового проекту потрібно дати коротку характеристику середовища розробки, а також описати всі об'єкти середовища, які були використані при написанні програми.

Наприклад:

2 СКЛАД СЕРЕДОВИЩА ПРОГРАМУВАННЯ DELPHI 7

Інтегроване середовище розробки (Integrated Development Environment — IDE) — це середовище, в якому є все необхідне для проектування, запуску та тестування прикладних додатків і в якому все підготоване для спрощення процесу розробки програм. Інтегроване середовище розробки інтегрує в себе редактор кодів, відгадчик, інструментальні панелі, редактор зображень, інструментарій баз даних – все, з чим приходить працювати. Ця інтеграція надає розробнику гармонійний набір інструментів. Крім того дозволяється розширювати меню середовища, включати в нього необхідні нам додаткові програми, в тому числі і власні.

Розгляну склад вікна середовища програмування Delphi . В верхній частині вікна знаходиться полоса головного меню. Її склад дещо відрізняється в залежності від версії.

Однією із найважливіших частин середовища Delphi являється вікно Редактора Кода.

Редактор Кода в Delphi 7 має дві сторінки: Code (код) и Diagram (діаграми).

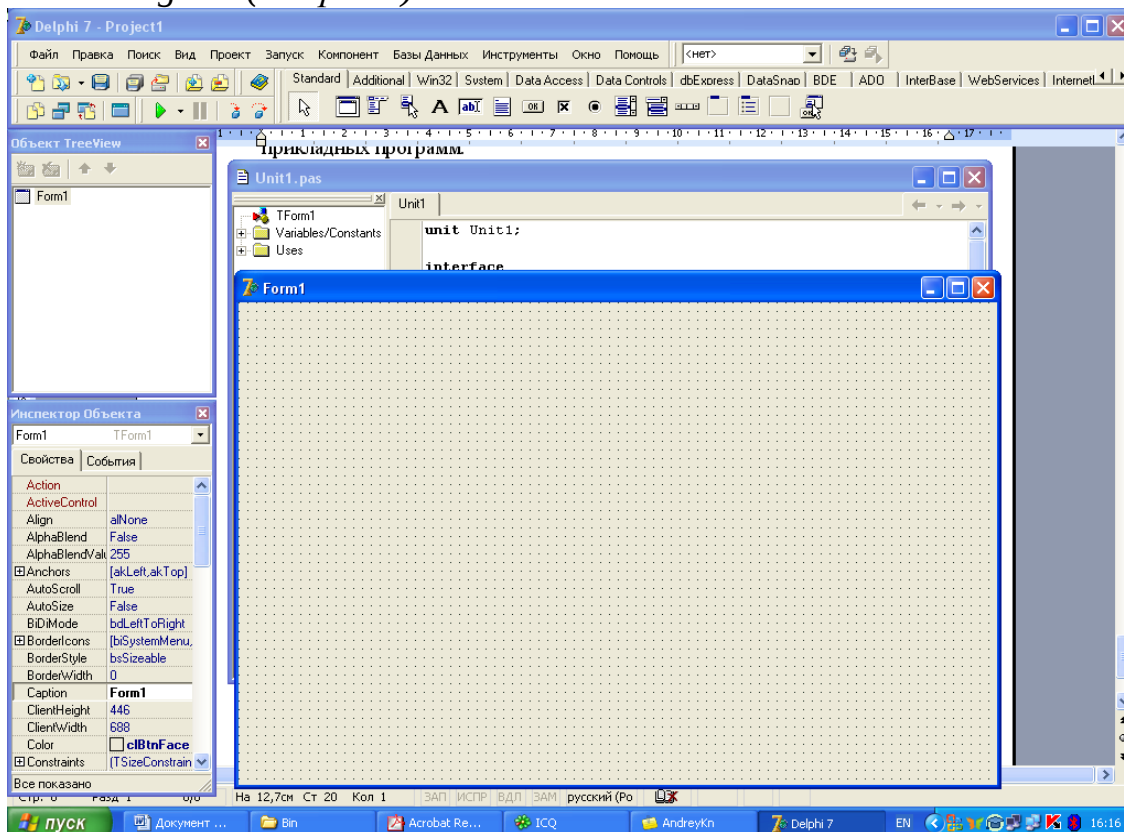


Рис 2.1. Вид середовища програмування Delphi 7

Кожна написана програму мовою Delphi складається із заголовка програми (program Project1;), поля використаних модулів Uses (наприклад, Uses Windows, Messages, SysUtils і т. д.), яке може не входити в загальну структуру, а також блоків виконання (починається з оператора begin і закінчується end.). Для проекту, котрий складається із двох форм, код проекту має вигляд:

```
program Project1; // Заголовок програми
uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1};
Unit2 in 'Unit2.pas' {Form2}; // модулі, котрі підключені до
проекту

{$R *.res}
```

```

begin
  Application.Initialize;           // ініціалізація додатку
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2); // створення форми/вікна
  Application.Run;                 // запуск і виконання
end.

```

Після того, як студент дав коротку характеристику середовища розробки, він повинен описати всі об'єкти, які були використані під час створення додатку (форми, кнопки, меню тощо), а також перераховані процедури та функції, які були створені з поясненням їх призначення. При цьому бажано додати декілька знімків екрану в режимі створення/редагування додатку.

Наприклад:

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 ОПИС МЕТОДІВ, КОТРІ БУЛИ ВИКОРИСТАНІ ПРИ РОЗРОБЦІ ПРОГРАМИ

Під час виконання курсового проекту я створив додаток, який включає в себе два модуля: Unit 1 і Unit 2. В Unit 1 знаходиться головна програма, котра викликає форму, що містить наступні об'єкти : Button (кнопка), Edit (поле вводу), ListBox (список строк), Label (надпис), MainMenu (головне меню програми), котре містить в собі підпункти N1 (пункт «Файл»), N2 (пункт «Розрахунок»), N5 (пункт «Вихід»), N7 (пункт «Про програму»), N3 та N6 (роздільники), OpenFileDialog та SaveDialog (діалогові вікна для відкриття та збереження файлу).

Перелік файлів, які входять в проект:

- Unit1.pas, Unit2.pas — коди модулів (pascal)
- Project1.dpr — код проекту (pascal)
- Project1.dfm, Project2.dfm — коди форм
- Project1.exe — скомпільований додаток

В Unit 1 описані такі процедури:

```

procedure N2Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure N7Click(Sender: TObject);
procedure N8Click(Sender: TObject);

```

Процедура N2Click викликається при натисненні N2 (кнопка «Розрахунок»).

Процедура N5Click викликається при натисненні N5 (кнопка “Вихід”).

Процедура N7Click викликається при натисненні N7 (кнопка “Про програму”).

Unit 2 викликає форму з даними про завдання на курсовий проект та виконавцем роботи

Після того, як студент описав всі компоненти програми, потрібно показати правила користування програмою, зробити знімки програми під час виконання та внести їх до пояснювальної записки. Крім того, бажано вказати необхідні мінімальні системні вимоги комп’ютера, які потрібні для запуску додатка.

У додатках студент повинен привести лістинг програми. Якщо додаток створений в середовищі Delphi, в додаток виноситься лише код, що міститься в файлах Project.dpr та Unit.pas (файл проекту та всі файли модулів). Коди форм в додаток виносити не потрібно, так як в пояснювальній записці повинні бути описані всі об’єкти, що містяться на формі, а також наведені знімки екрану. Кожен додаток повинен починатися з нової сторінки.

Приклад оформлення додатків:

ДОДАТКИ

Додаток 1

Лістинг файла Unit2.pas

```
unit unit2;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, Buttons;

type
  TfrmPurpose = class(TForm)
    Memo1: TMemo;
    BitBtn1: TBitBtn;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var  
    frmPurpose: TfrmPurpose;  
  
implementation  
  
{$R *.dfm}  
  
end.
```

В кінці пояснювальної записки студент повинен оформити графічну частину курсового проекту. Для цього на аркушах формату А3 студент повинен подати наступні об'єкти:

- Загальна схема роботи програми;
- Блок-схеми роботи кожного модуля;
- Лістинг всіх файлів, що входять до складу проекту.

Бажано скомпонувати ці модулі таким чином, щоб блок-схема, схема роботи та лістинг файлу, що відносяться до кожного із модулів, містилися на одному аркуші (наприклад, на першому аркуші графічної частини – схема роботи програми та лістинг файлу Project1.dpr, на другому – блок-схема та лістинг файлу Unit1.pas, на третьому – знімок вікна програми та лістинг файлу Project1.dfm), але дозволяється спочатку надати всі алгоритми (блок-схеми) роботи програми та модулів, а потім лістинги. При оформленні потрібно дотримуватися наступних вимог:

- Орієнтація аркуша – альбомна;
- Оформлення основного надпису – згідно вимог (штамп 40 мм, на кожному із аркушів вказано назву графічної частини, нумерація для графічної частини починається з початку);
- Аркуші підшиваються після пояснювальної записки;

4 ОРІЄНТОВНЕ ЗАВДАННЯ НА КУРСОВЕ ПРОЕКТУВАННЯ

Під час виконання курсового проекту студент розробляє програмний код та оформлює пояснювальну записку згідно завдання. В переліку вказано теми курсового проекту, вхідні дані для проектування та критерії оцінювання виконаного проекту.

Варіант 1.

Тема: Робота зі списками.

Вхідні дані для проектування: Створити лінійний список, елементами якого являється запис «Квартира» з полями «Вулиця», «Номер будинку», «Номер квартири», «Загальна площа», «Кількість жителів». Організувати можливість перегляду списку та додання нового елемента.

Критерії оцінювання:

"задовільно" – створити процедури перегляду елементів списку, додання нового елемента та збереження списку.

"добре" – створити процедури перегляду елементів списку, додання нового елемента, видалення елемента та збереження списку.

"відмінно" – створити процедури перегляду елементів списку, додання нового елемента, видалення елемента, пошуку елемента та збереження списку.

Варіант 2.

Тема: Робота зі списками.

Вхідні дані для проектування: Створити лінійний список, елементами якого являється запис «Автомобіль» з полями «Марка», «Номер», «Рік випуску», «Прізвище власника». Організувати можливість перегляду списку та додання нового елемента.

Критерії оцінювання:

"задовільно" – створити процедури перегляду елементів списку, додання нового елемента та збереження списку.

"добре" – створити процедури перегляду елементів списку, додання нового елемента, видалення елемента та збереження списку.

"відмінно" – створити процедури перегляду елементів списку, додання нового елемента, видалення елемента, пошуку елемента та збереження списку.

Варіант 3.

Тема: Робота зі списками.

Вхідні дані для проектування: Створити лінійний список, елементами якого являється запис «Людина» з полями «Прізвище», «Ім'я», «Побатькові», «Дата народження», «Стать». Організувати можливість перегляду списку та додавання нового елемента.

Критерії оцінювання:

"задовільно" – створити процедури перегляду елементів списку, додавання нового елемента та збереження списку.

"добре" – створити процедури перегляду елементів списку, додавання нового елемента, видалення елемента та збереження списку.

"відмінно" – створити процедури перегляду елементів списку, додавання нового елемента, видалення елемента, пошуку елемента та збереження списку.

Варіант 4.

Тема: Розробка програми тестування.

Вхідні дані для проектування: Розробити програму тестування знань по темі «Мова програмування Pascal». Програма повинна містити щонайменше 10 запитань, на кожне – кілька варіантів відповідей, з яких лише одна правильна. В кінці повинна виводитися кількість правильних відповідей та оцінка тестування.

"задовільно" – всі питання тестів записані лише на формі програми тестування, виведення результатів тестування проводиться за допомогою простого вікна повідомлень.

"добре" – перед початком тестування передбачити можливість реєстрації, всі питання тестів записані лише на формі програми тестування, виведення результатів тестування проводиться за допомогою простого вікна повідомлень і дублюється в спеціальний файл звіту.

"відмінно" – перед початком тестування передбачити можливість реєстрації, всі питання тестів зберігаються в окремому текстовому файлі, виведення результатів тестування проводиться за допомогою простого вікна повідомлень або нової форми і дублюється в спеціальний файл звіту.

Варіант 5.

Тема: Розробка програми тестування.

Вхідні дані для проектування: Розробити програму тестування знань по темі «Ключові слова мови програмування Pascal». Програма повинна містити щонайменше 10 запитань, на кожне – кілька варіантів відповідей, з яких лише одна правильна. В кінці повинна виводитися кількість правильних відповідей та оцінка тестування.

Критерії оцінювання:

"задовільно"— програма має лише один варіант тестів, всі питання тестів записані лише на формі програми тестування, виведення результатів тестування проводиться за допомогою простого вікна повідомлень.

"добре" – програма має лише декілька варіантів тестів (щонайменше два), з яких користувач обирає потрібний, всі питання тестів записані лише на формі програми тестування, виведення результатів тестування проводиться за допомогою простого вікна повідомлень.

"відмінно" – програма має лише декілька варіантів тестів (щонайменше два), з яких користувач обирає потрібний, всі питання тестів зберігаються в окремому текстовому файлі, виведення результатів тестування проводиться за допомогою простого вікна повідомлень або нової форми.

Варіант 6.

Тема: Розробка програми тестування.

Вхідні дані для проектування: Розробити програму тестування знань по темі «Властивості форми». Програма повинна містити щонайменше 10 запитань, на кожне – кілька варіантів відповідей, з яких лише одна правильна. В кінці повинна виводитися кількість правильних відповідей та оцінка тестування.

Критерії оцінювання:

"задовільно" – програма має лише один варіант тестів, всі питання тестів записані лише на формі програми тестування, виведення результатів тестування проводиться за допомогою простого вікна повідомлень.

"добре" – програма має лише декілька варіантів тестів (щонайменше два), з яких користувач обирає потрібний, всі питання тестів записані лише на формі програми тестування, виведення результатів тестування проводиться за допомогою простого вікна повідомлень.

"відмінно" – програма має лише декілька варіантів тестів (щонайменше два), з яких користувач обирає потрібний, всі питання тестів зберігаються в окремому текстовому файлі, виведення результатів тестування проводиться за допомогою простого вікна повідомлень або нової форми.

Варіант 7.

Тема: Розробка програми тестування.

Вхідні дані для проектування: Розробити програму тестування знань по темі «Візуальні компоненти середовища Delphi». Програма повинна містити щонайменше 10 запитань, на кожне – кілька варіантів відповідей, з яких лише одна правильна. В кінці повинна виводитися кількість правильних відповідей та оцінка тестування.

Критерії оцінювання:

"задовільно" – всі питання тестів записані лише на формі програми тестування, виведення результатів тестування проводиться за допомогою простого вікна повідомлень.

"добре" – перед початком тестування передбачити можливість реєстрації, всі питання тестів записані лише на формі програми тестування, виведення результатів тестування проводиться за допомогою простого вікна повідомлень і дублюється в спеціальний файл звіту.

"відмінно" – перед початком тестування передбачити можливість реєстрації, всі питання тестів зберігаються в окремому текстовому файлі, виведення результатів тестування проводиться за допомогою простого вікна повідомлень або нової форми і дублюється в спеціальний файл звіту.

Варіант 8.

Тема: Робота з таблицями.

Вхідні дані для проектування: За допомогою об'єкта StringGrid побудувати зведену таблицю семестрових оцінок групи, створити процедуру (чи функцію) підрахунку середнього балу успішності кожного студента, відсортувати по середньому балу. Передбачити можливість друку отриманих результатів у вигляді таблиці.

Критерії оцінювання:

"задовільно" – дані вводяться в таблицю StringGrid вручну або програмно, обробка даних проводиться лише на формі.

"добре" – дані беруться із зовнішнього типізованого файлу або файлу Excel, обробка та друк даних проводяться з форми.

"відмінно" – дані беруться із зовнішнього типізованого файлу або файлу Excel, обробка та друк даних проводяться з форми, передбачити можливість перезапису даних та дозапису результатів у вихідний файл.

Варіант 9.

Тема: Робота з таблицями.

Вхідні дані для проектування: За допомогою об'єкта StringGrid побудувати зведену таблицю семестрових оцінок групи, створити процедуру (чи функцію) підрахунку середнього балу успішності кожного студента, відсортувати по середньому балу. Передбачити можливість друку отриманих результатів у вигляді таблиці.

Критерії оцінювання:

"задовільно" – дані беруться зі вказаних комірок таблиці, підрахунок виконується лише на формі без запису результату у файл.

"добре" – дані беруться зі вказаних комірок таблиці, передбачена можливість дозапису результатів у вихідний файл.

"відмінно" – дані беруться зі вказаних комірок таблиці, передбачена можливість перезапису даних та дозапису результатів у вихідний файл.

Варіант 10.

Тема: Управління додатком.

Вхідні дані для проектування: **Тема:** Управління додатком.

Вхідні дані для проектування: Написати програму, яка буде відслідковувати натиснення на кнопки клавіатури та записувати код або значення кожної натисненої клавіші в окремий файл.

Критерії оцінювання:

"задовільно" – проводити дозапис коду або значення кнопки основної групи клавіш в заздалегідь визначений файл звіту.

"добре" – файл повинен створюватися в той момент, коли користувач запустить додаток і в назві містити дату і час запуску.

"відмінно" – в створений при запуску файл звіту додатково заносити дані про натиснення на функціональні клавіші.

Варіант 11.

Тема: Управління додатком.

Вхідні дані для проектування: Створити форму, на якій буде міститися хоч 1 об'єкт. Створити програму, яка буде відслідковувати координати вказівника миші і тоді, коли вказівник буде наближуватися до заданого об'єкта, змінювати його координати, переміщуючи його в довільне місце форми, але не перекриваючи інших об'єктів та не виходячи за межі.

Критерії оцінювання:

"задовільно" – заданий об'єкт довільно змінює своє положення випадковим чином. Як об'єкт взяти будь-який стандартний компонент.

"добре" – об'єкт плавно змінює свої координати в напрямку, протилежному вказівнику. Як об'єкт взяти будь-який стандартний компонент.

"відмінно" – об'єкт плавно змінює свої координати в напрямку, протилежному вказівнику. Як об'єкт взяти рисунок з деякого визначеного користувачем файлу, поміщеного в поле Image.

Варіант 12.

Тема: Управління додатком.

Вхідні дані для проектування: Створити програму, яка буде відслідковувати координати вказівника миші записувати всі події миші в визначений файл звіту.

Критерії оцінювання:

"задовільно" – на формі у визначеній позиції (наприклад, за допомогою надпису) виводити координати вказівника, записувати у файл звіту координати вказівника при події OnClick.

"добре" – на формі на позиції вказівника миші виводити його координати, записувати у файл звіту координати вказівника при події OnClick.

"відмінно" – на формі на позиції вказівника миші виводити його координати, записувати у файл звіту координати вказівника, дату та час події OnClick.

Варіант 13.

Тема: Робота з файлами.

Вхідні дані для проектування: Створити програму – аналог програми "Провідник", яка виводить на екран в вигляді таблиці перелік файлів у вказаній користувачем папці та їх атрибути.

Критерії оцінювання:

"задовільно" – створити процедури перегляду вмісту вказаної користувачем папки та друку вмісту у вигляді списку.

"добре" – створити процедури перегляду вмісту вказаної користувачем папки з виводом атрибутів файлів та друку вмісту у вигляді списку.

"відмінно" – створити процедури перегляду вмісту вказаної користувачем папки з виводом атрибутів файлів та друку вмісту у вигляді таблиці.

Варіант 14.

Тема: Розробка прикладних програм.

Вхідні дані для проектування: Розробити аналог калькулятора. Програма повинна мати зручний дизайн, можливість вводу даних як з клавіатури, так і за допомогою кнопок, розташованих на формі та виконувати усі основні операції.

Критерії оцінювання:

"задовільно" – введення даних проводиться як з клавіатури, так і за допомогою кнопок на формі, перевірка коректності введених даних не проводиться, виконуються лише 4 основних арифметичних операції.

"добре" – введення даних проводиться як з клавіатури, так і за допомогою кнопок на формі, проводиться перевірка коректності введених даних, виконуються лише 4 основних арифметичних операції.

"відмінно" – введення даних проводиться як з клавіатури, так і за допомогою кнопок на формі, за допомогою подій клавіатури унеможлиблюється введення некоректних даних.

Варіант 15.

Тема: Розробка прикладних програм.

Вхідні дані для проектування: Створити програму, за допомогою якої можна переводити числа з десяткової системи числення в двійкову та навпаки. Заборонено користуватися стандартними функціями, всі процедури прописує розробник.

Критерії оцінювання:

"задовільно" – передбачити можливість переведення з однієї системи в іншу без перевірки коректності введених даних.

"добре" – провести перевірку коректності введених даних, у випадку некоректного введення видати повідомлення, в протилежному – перевести запис числа з однієї системи в іншу.

"відмінно" – заблокувати можливість введення некоректних даних до полів вводу та перевести запис числа з однієї системи в іншу.

Варіант 16.

Тема: Розробка прикладних програм.

Вхідні дані для проектування: Створити програму, за допомогою якої можна переводити числа з десяткової системи числення в вісімкову та навпаки. Заборонено користуватися стандартними функціями, всі процедури прописує розробник.

Критерії оцінювання:

"задовільно" – передбачити можливість переведення з однієї системи в іншу без перевірки коректності введених даних.

"добре" – провести перевірку коректності введених даних, у випадку некоректного введення видати повідомлення, в протилежному – перевести запис числа з однієї системи в іншу.

"відмінно" – заблокувати можливість введення некоректних даних до полів вводу та перевести запис числа з однієї системи в іншу.

Варіант 17.

Тема: Розробка прикладних програм.

Вхідні дані для проектування: Створити програму, за допомогою якої можна переводити числа з десяткової системи числення в шістнадцяткову та навпаки. Заборонено користуватися стандартними функціями, всі процедури прописує розробник.

Критерії оцінювання:

"задовільно" – передбачити можливість переведення з однієї системи в іншу без перевірки коректності введених даних.

"добре" – провести перевірку коректності введених даних, у випадку некоректного введення видати повідомлення, в протилежному – перевести запис числа з однієї системи в іншу.

"відмінно" – заблокувати можливість введення некоректних даних до полів вводу та перевести запис числа з однієї системи в іншу.

Варіант 18.

Тема: Програма для кодування даних.

Вхідні дані для проектування: Створити додаток, за допомогою якого можна закодувати за деяким принципом вміст текстового файлу та провести обернену операцію. Передбачити можливість перегляду вмісту обох файлів через об'єкти форми (наприклад, Метод) за вимогою користувача.

Критерії оцінювання:

"задовільно" – закодувати інформацію методом простої перестановки символів.

"добре" – методом заміни символів за певними правилами.

"відмінно" – комбінація методів заміни та перестановки.

Варіант 19.

Тема: Програма для кодування даних.

Вхідні дані для проектування: Створити додаток, за допомогою якого можна закодувати за деяким принципом введені текстові та провести обернену операцію. Передбачити можливість завантаження та збереження даних за вимогою користувача.

Критерії оцінювання:

"задовільно" – закодувати інформацію методом простої перестановки символів.

"добре" – методом заміни символів за певними правилами.

"відмінно" – комбінація методів заміни та перестановки.

Варіант 20.

Тема: Розробка прикладної програми.

Вхідні дані для проектування: Створити найпростіший ігровий додаток на вибір студента.

Критерії оцінювання:

"задовільно" – управління здійснюється за допомогою кнопок, доданих на форму.

"добре" – управління здійснюється за допомогою відслідковування подій клавіатури.

"відмінно" – управління здійснюється за допомогою відслідковування подій миші.

ЛІТЕРАТУРА

1. ГОСТ 19. 404 – 79. Пояснительная записка. Требования к содержанию и оформлению.
2. ГОСТ 7.1-84. Библиографическое описание документа. Общие требования и правила составления.
3. ДСТУ 3582-97. Інформація та документація. Скорочення слів в українській мові в бібліографічному описі. Загальні вимоги та правила.
4. ГОСТ 7.12-93. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила.
5. Основы программирования в Delphi 7. – СПб.:БХВ – Петербург, 2006.- 608 с.: ил.
6. Turbo Pascal.- СПб.: БХВ – Петербург, 2006.- 1056 с.
7. Голощук Р.О. та ін. Алгоритми і структури даних. Л.: "Магнолія", 2010.
8. Кузнецов Ю.Н. и др. Математическое программирование. М.: Высшая школа, 2001.
9. Кетков Ю. и др. Практика по программированию: Visual Basic, C++, Builder, Delphi.
10. Бобровський С. Delphi 7. Навчальний курс.

ДОДАТКИ

Додаток 1

Приклад оформлення змісту роботи

ЗМІСТ

ВСТУП	4
1 ПОСТАНОВКА ЗАВДАННЯ	5
1.1 АНАЛІЗ ЗАВДАННЯ, ВИБІР МЕТОДІВ ОБРОБКИ ДАНИХ	6
1.2 СТВОРЕННЯ АЛГОРИТМУ РОБОТИ ПРОГРАМИ	10
2 СКЛАД СЕРЕДОВИЩА ПРОГРАМУВАННЯ	12
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	17
3.1 ОПИС МЕТОДІВ, КОТРІ БУЛИ ВИКОРИСТАНІ ПРИ РОЗРОБЦІ ПРОГРАМИ	22
3.2 РОБОТА З ПРОЕКТОМ	27
4 ОХОРОНА ПРАЦІ І ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ	30
ВИСНОВОК	33
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	34
ДОДАТКИ	35

					5.123.1.20.____ - ПЗ			
Изм.	Арк.	№ документа	Підпис	Дата				
Розробив		Іванов І.І.			Пояснювальна записка	Літера	Аркуш	Аркушів
Перевірив						I	3	38
Н. контр.						ХДПК група 47-КМ		
Затвердив								

2 СКЛАД СЕРЕДОВИЩА ПРОГРАМУВАННЯ DELPHI 7

Інтегроване середовище розробки (Integrated Development Environment — IDE) — це середовище, в якому є все необхідне для створення та проектування, запуску та тестування прикладних додатків і в якому все підготоване для спрощення процесу розробки програм. Інтегроване середовище розробки інтегрує в себе редактор кодів, відладчик, інструментальні панелі, редактор зображень, інструментарій баз даних – все, з чим приходиться працювати. Ця інтеграція надає розробнику гармонійний набір інструментів. Крім того дозволяється розширювати меню середовища, включати в нього необхідні нам додаткові програми, в тому числі і власні.

Розгляну склад вікна середовища програмування Delphi . В верхній частині вікна знаходиться полоса головного меню. Її склад дещо відрізняється в залежності від версії.

Однією із найважливіших частин середовища Delphi являється вікно Редактора Кода.

Редактор Кода в Delphi 7 має дві сторінки: Code (код) и Diagram (діаграми).

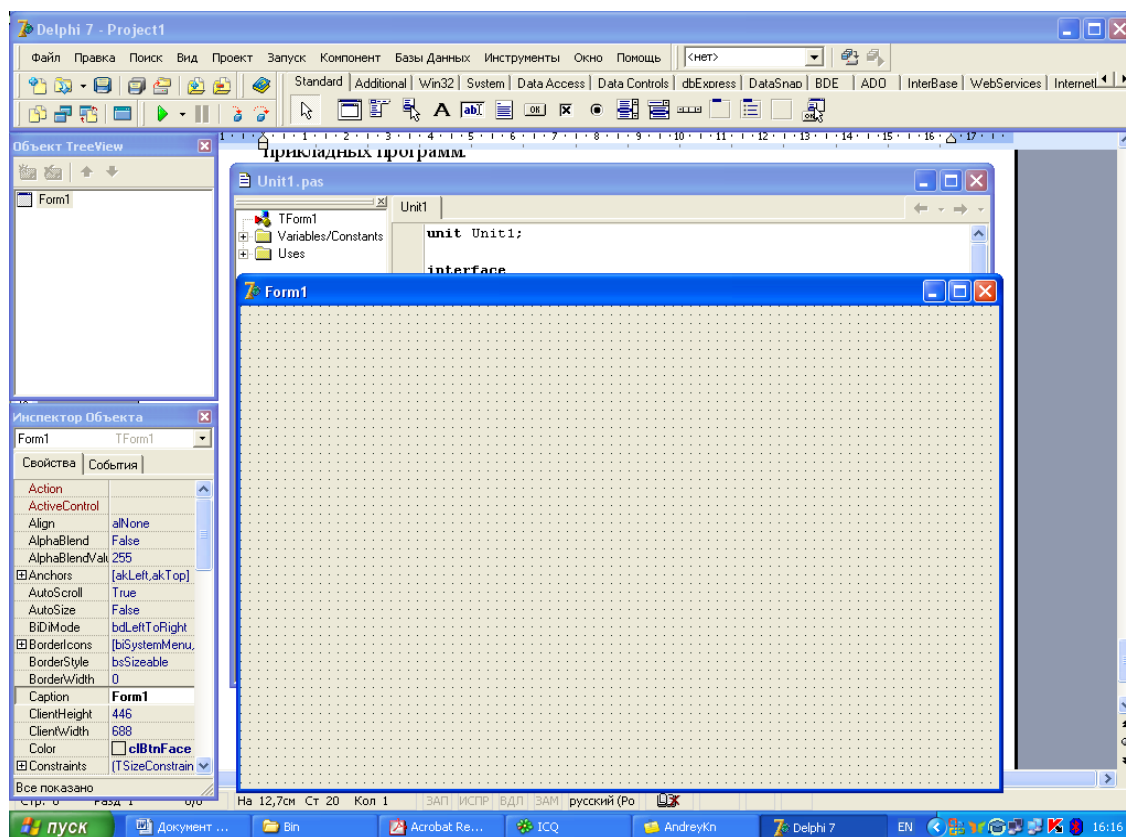


Рис 2.1. Вид середовища програмування Delphi 7

СПИСОК ЛІТЕРАТУРИ

1. Антоненко І.В. Основи програмування в Delphi 7. Л.: "Магнолія", 2014.
2. Глушаков С.В. Turbo Pascal.- СПб.: БХВ – Петербург, 2006.- 1056 с.
3. Голощук Р.О. та ін. Алгоритми і структури даних. Л.: "Магнолія", 2015.
4. Кузнецов Ю.Н. и др. Математическое программирование. К.: Высшая школа, 2001.
5. Кетков Ю. и др. Практика по программированию: Visual Basic, C++, Builder, Delphi. К.: Высшая школа, 2015

