

*Т.П.Караванова, В.П.Костюков*

**Програма курсу за вибором  
«Основи алгоритмізації та  
програмування»**

**для організації профільного навчання  
у старших класах загальноосвітніх  
навчальних закладів**

**2011**

## ***Пояснювальна записка***

Інформатика на сьогоднішній день є одним із засобів формування не тільки освітнього, а й розвиваючого та інтелектуального потенціалу особистості. У процесі вивчення інформатики основні завдання курсу за вибором «Основи алгоритмізації та об'єктно-орієнтованого програмування» мають на меті виявлення та розвиток в учнів логічних здібностей, підготовки їх до участі в інтелектуальних змаганнях та наукових дискусіях, формування в них стійкого інтересу до алгоритмізації і пов'язаної з нею професійної діяльності, підготовки до навчання у вищих навчальних закладах.

### ***Програма ставить за мету:***

- розвиток логічного, аналітичного мислення та основних видів розумової діяльності: уміння використовувати індукцію, дедукцію, аналіз, синтез, робити висновки, узагальнення;
- формування теоретичної бази знань учнів щодо процесів перетворення, передавання та використання інформації, розкриття значення інформаційних процесів у формуванні сучасної системно-інформаційної картини світу, розкриття ролі інформаційних технологій в розвитку сучасного суспільства;
- розвиток уміння розв'язувати змістовні задачі різного рівня складності, користуючись відомими теоретичними положеннями, математичним апаратом, літературою та комп'ютерною технікою;
- розуміння того, що при розв'язуванні задач за допомогою комп'ютера первинним є розробка алгоритму поставленої задачі, а мова програмування – це лише інструмент для його реалізації;
- доведення вивчення алгоритмізації до творчого рівня;
- бачення учнями можливостей використання набутих знань у їх майбутній професії;
- інтеграцію інформатики з іншими предметами, що викладаються в навчальних закладах.

*До теоретичної бази знань відносяться:*

- основи алгоритмізації та програмування;
- постановка задач і побудова відповідних інформаційних (зокрема, математичних) моделей, загальні принципи розв'язування задач за допомогою комп'ютера з використанням програмного забезпечення загального та навчального призначення.

*До практичних навичок відносяться:*

- навички аналізу відомих методів побудови алгоритмів та визначення найоптимальніших з них для розв'язування конкретної задачі;
- навички тестування складених алгоритмів;
- навички роботи з інтегрованим середовищем візуального програмування;
- навички техніки програмування.

Мета курсу досягається через практичне оволодіння учнями навичками роботи з основними складовими сучасного програмного забезпечення комп'ютерів, з основами технології розв'язування задач за допомогою комп'ютера, починаючи від їх постановки й побудови відповідних інформаційних моделей і завершуючи інтерпретацією результатів, отриманих за допомогою комп'ютера.

Однією з головних ідей, покладених у розробку програми, є *стимулювання самостійної роботи* учнів шляхом виконання власних проектів протягом вивчення всіх тем курсу. Це в першу чергу спонукає до *розвитку їх творчого креативного мислення* під час опанування даного курсу.

На передній план даної програми з методичної точки зору виноситься *мотиваційна діяльність кожного учня*. А саме вироблення стилю та культури програмування, створення власної бібліотеки навчальних програм, проектна діяльність, виховання алгоритмічної культури.

Програма курсу розрахована на вивчення основ алгоритмізації та об'єктно-орієнтованого програмування за варіантом постійного

використання комп'ютерів. Вивчення курсу передбачає 105 годин, а саме 100 навчальних годин та 5 резервних годин протягом одного навчального року по 3 години на тиждень.

*Основною формою навчальних занять* залишаються уроки різних типів: вивчення нового навчального матеріалу, удосконалення знань та формування умінь при розв'язуванні задач, узагальнення та систематизація знань, контроль та корекція знань. Рекомендується використовувати такі форми організації навчання як індивідуальні, роботу в малих та великих групах тощо. Під час вивчення курсу передбачаються такі практичні форми занять: уроки розв'язування задач, лабораторні роботи, роботи над проектними задачами. Особливо слід зазначити відмінність між практичними та лабораторними роботами, які передбачені у програмі як одна із форм звітності учнів за набуті знання з окремих тем. Якщо практична робота може виконуватися учнями як індивідуально, так і у групах, відповідно до обраної вчителем методики, і передбачає виконання конкретного однакового для всіх учнів завдання, то лабораторні роботи рекомендуються до виконання учнями самостійно. Вони передбачають ведення дослідження у зошиті з окремої теми по кожній лабораторній роботі із зазначенням постановки задачі, опису розробленого алгоритму, підібраних власних тестів, результатів тестування, аналітичного дослідження отриманих результатів, порівняння з результатами інших альтернативних методів розв'язання поставленої задачі, остаточних висновків. Результатом виконання лабораторної роботи є її захист у вчителя. Такий захист на вибір вчителя може бути проведений індивідуально з окремими учнями, або у груповому інтерактивному режимі. У програмі лабораторні роботи передбачені лише із тих тем, які дозволяють учням провести відповідну дослідницьку роботу.

Бібліотека навчальних програм створюється учнями під час вивчення основ алгоритмізації та програмування і по завершенні всього курсу ці наробки використовуються учнями під час роботи над завершальним

проектом. Теми такого глобального проекту пропонуються самими учнями та узгоджуються з вчителем, або ж є результатом співпраці з іншими вчителями-предметниками, а також сторонніми організаціями, фірмами тощо.

*Критерії оцінювання навчальних досягнень* визначаються через виконання учнями практичних та лабораторних робіт, захист власних проектних робіт з різних тем, проведення учителем тематичного оцінювання знань учнів як підсумковий етап з окремих тем або групи послідовних тем.

*Початковий рівень навчальних досягнень* визначається у разі репродуктивної діяльності учня, а саме визначення ним основних понять теми, повторення прикладів, що наводилися вчителем, виконання практичних завдань, що розглядалися на уроці під час ознайомлення з новим матеріалом, виконання лабораторних робіт у повній відповідності з вказівками вчителя та під його керівництвом. При розробці проектів учень бере участь у групах за визначеною вчителем темою.

*Середній рівень навчальних досягнень* передбачає самостійне виконання учнем завдань, подібних до тих, що розглядалися вчителем під час подання нового матеріалу. Учень також орієнтується в теоретичному матеріалі, відповідає на запитання вчителя не лише в тій послідовності, в якій подавався новий матеріал, виконує практичні та лабораторні роботи частково самостійно, використовуючи тести, запропоновані вчителем. Розробку проектів учень виконує під керівництвом та постійним контролем вчителя за визначеною ним темою.

*Достатній рівень навчальних досягнень* демонструють учні, які орієнтуються у навчальному матеріалі нової теми, відповідаючи на запитання вчителя, самостійно виконують практичні та лабораторні роботи, підбираючи для цього власні тести, ґрунтовно аналізують отримані результати. Під час роботи над проектами учні працюють самостійно під керівництвом вчителя.

*Високий рівень навчальних досягнень* учнів визначається у разі вільної орієнтації у новому навчальному матеріалі з можливою участю при його викладенні вчителем під час уроку, розуміння взаємопов'язаності різних тем всієї програми, самостійного творчого виконання практичних та лабораторних робіт, аналітичного підходу до аналізу отриманих результатів, самостійної проектної роботи, участі в інтелектуальних змаганнях з інформатики (олімпіади, конкурси, конференції, турніри, захисти наукових робіт тощо).

Дана програма розроблена для використання у 10-му класі і базується на знаннях учнів, які вони отримали у курсі інформатики у 9-му класі.

Програма розроблена таким чином, щоб дати можливість використовувати її як у варіанті 70-годинного річного навантаження (66 навчальних годин, 4 резервних години, 2 години на тиждень) та 105-годинного річного навантаження (100 навчальних годин, 5 резервних годин, 3 години на тиждень). Теми, помічені у програмі «\*», у 70-годинному варіанті не розглядаються зовсім. Для тем, які мають різну кількість годин, у дужках вказані години для 70-годинному варіанту. Різниця в навчальному часі для таких тем забезпечується завдяки виконанню різної кількості вправ, практичних і лабораторних робіт. Практичні і лабораторні роботи, що мають виконуватися лише в повному варіанті курсу, також позначено символом «\*». У разі, коли кількість годин теми жодним чином не відмічена, то це означає, що в обох варіантах вона залишається незмінною.

Тематика даної програми розроблена таким чином, щоб в учнів за бажанням була можливість продовжити профільне вивчення алгоритмізації та програмування в 11 класі.

## Зміст навчального матеріалу та вимоги до навчальних досягнень учнів

Зміст навчального матеріалу	Навчальні досягнення учнів
<b>Розділ «Основи алгоритмізації та об'єктно-орієнтованого програмування», 100+5 (66+4) годин</b>	
<i>I семестр (48 год. (46+2 рез.), 3 год. на тиждень )                      Або (32 год. (30+2 рез.), 2 год. на тиждень )                      Теми, помічені «*», у другому варіанті програми не розглядаються</i>	
<p><b>Тема 1: «Основні поняття алгоритмізації», 4 години</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>поняття алгоритму; властивості алгоритмів; способи представлення алгоритмів; базові алгоритмічні структури; типи алгоритмів; виконавець та система команд виконавця; формальне виконання алгоритму; аргументи, результати, проміжні величини</p> <p><b>Практичні роботи:</b></p> <p>«Способи представлення алгоритмів»</p> <p>«Базові алгоритмічні структури. Типи алгоритмів»</p>	<p><i><b>називає:</b></i> основні поняття алгоритмізації, властивості алгоритмів, способи представлення алгоритмів, базові алгоритмічні структури, типи алгоритмів, аргументи, результати, проміжні величини</p> <p><i><b>наводить:</b></i> власні приклади алгоритмів та виконавців алгоритмів, аргументів, результатів, проміжних величин</p> <p><i><b>розпізнає:</b></i> відмінність між різними базовими алгоритмічними структурами та типами алгоритмів, аргументами, результатами, проміжними величинами</p> <p><i><b>характеризує:</b></i> задані алгоритми за типами, способами представлення</p> <p><i><b>описує:</b></i> задані алгоритми різними способами</p>

**пояснює:** сутність поняття виконавця алгоритму та системи команд виконавця, сутність поняття формального виконання алгоритму, виконання дій заданого алгоритму, поняття аргументу, результату та проміжної величини

**формулює:** означення алгоритму, базових алгоритмічних структур, характерні ознаки різних типів алгоритмів, аргументів, результатів та проміжних величин

**обґрунтовує:** наявність тих чи інших базових алгоритмічних структур у заданому алгоритмі, належність конкретного алгоритму до відповідного типу, коректність дій при виконанні заданого алгоритму, наявність у заданому алгоритмі аргументів, результатів та проміжних величин

**порівнює:** різні способи представлення алгоритмів, різні алгоритми за типами, змінні як аргументи, результати та проміжні величини

**класифікує:** алгоритми за способами представлення та за типами, змінні як аргументи, результати та проміжні

величини

**аналізує:** належність заданих алгоритмів до тих чи інших типів, визначення змінних як аргументів, результатів або проміжних величин, власні помилки і помилки інших учнів при виконанні завдань та під час відповідей на запитання учителя

**оцінює:** складність виконуваних завдань, коректність власних відповідей та відповідей інших учнів

**висловлює судження:** щодо належності заданої послідовності дій поняттю алгоритму та наявності у ній всіх його властивостей, наявності у заданих алгоритмах тих чи інших базових алгоритмічних структур, належності заданих алгоритмів тим чи іншим типам, належності змінних заданого алгоритму до аргументів, результатів або проміжних величин

**розв'язує:** завдання по представленню заданих алгоритмів різними способами, визначенню у заданих алгоритмах аргументів, результатів та проміжних величин

	<p><i>спостерігає</i>: за виконанням завдань іншими учнями, аналізуючи їх відповіді</p> <p><i>використовує</i>: власні знання і рекомендації учителя для виконання завдань</p> <p><i>дотримується правил</i>: опису алгоритмів різними способами з використанням відповідних базових структур</p>
<p><b>Тема 2: «Моделі та моделювання», 2 години (*)</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>поняття моделі та моделювання; класифікація моделей; інформаційна модель; математична модель як різновид інформаційної моделі; етапи розв’язування задач на комп’ютері</p> <p><b>Практичні роботи:</b></p> <p>«Побудова інформаційної та математичної моделі»</p>	<p><i>називає</i>: класи моделей, етапи розв’язування задач на комп’ютері</p> <p><i>наводить</i>: приклади моделей</p> <p><i>розпізнає</i>: класи моделей</p> <p><i>характеризує</i>: класи моделей, окремі етапи розв’язування задач на комп’ютері</p> <p><i>описує</i>: моделі за класами, інформаційну модель, математичну модель, запропонованих задач</p> <p><i>пояснює</i>: смисл моделювання та його прикладне значення</p> <p><i>формулює</i>: поняття моделі, моделювання, інформаційної моделі, математичної моделі</p> <p><i>обґрунтовує</i>: належність конкретних моделей до</p>

відповідних класів, послідовність етапів розв'язування задач на комп'ютері, необхідність створення інформаційної та математичної моделі задачі

**порівнює:** різні класи моделей

**класифікує:** конкретні приклади моделей

**аналізує:** належність конкретних моделей до відповідних класів, необхідність створення інформаційної та математичної моделі задачі для наступного її розв'язування, доцільність певних етапів розв'язування задачі на комп'ютері для кожного конкретного прикладу

**оцінює:** складність запитань вчителя, повноту відповідей як власних так й інших учнів класу

**висловлює судження:** щодо місця та ролі моделей та моделювання у різних сферах діяльності людини, доцільності розбиття моделей на класи, логічної послідовності поетапного розв'язування задач на комп'ютері

**розв'язує:** запропоновані вчителем завдання щодо

	<p>визначення класу заданої моделі, створення інформаційної та математичної моделі</p> <p><i>спостерігає</i>: за логікою послідовності викладення матеріалу, за коректністю відповідей учнів</p> <p><i>використовує</i>: набуті знання і навички для виконання поставлених вчителем завдань як на уроках інформатики, так і на уроках з інших предметів</p> <p><i>дотримується правил</i>: створення інформаційної та математичної моделі, визначення етапів розв'язування задач на комп'ютері</p>
<p><b>Тема 3: «Мови програмування», 2 години (*)</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>мова програмування як один із способів представлення алгоритму; класифікація мов програмування; елементи мови програмування</p>	<p><i>називає</i>: різні мови програмування, класи мов програмування</p> <p><i>наводить</i>: приклади мов програмування, перелік класів мов програмування</p> <p>елементів мови програмування</p> <p><i>розпізнає</i>: різні мови програмування та їх призначення, до якого класу відноситься конкретна мова програмування; елементи мови програмування</p>

**характеризує:** мову програмування як один із способів представлення алгоритму, особливості різних класів мов програмування, особливості мови програмування, що вивчається

**описує:** призначення мов програмування, ознаки різних класів мов програмування

**пояснює:** призначення окремих мов програмування, ознаки окремих класів мов програмування

**формулює:** поняття класів мов програмування

**обґрунтовує:** наявність великої кількості різних мов програмування, ознаки окремих класів мов програмування, необхідність розробки різних середовищ програмування, призначення середовищ програмування

**порівнює:** особливості різних мов програмування

**класифікує:** мови програмування

**аналізує:** значення коректного вибору мови програмування для розв'язання поставленої задачі

**оцінює:** ефективність вибору мови програмування при

	<p>розв'язуванні поставленої задачі</p> <p><i>висловлює судження</i>: щодо необхідності створення різних мов програмування, необхідності та раціональності поділу мов програмування на класи</p> <p><i>розв'язує</i>: завдання щодо ідентифікації тексту програми відповідно до мови програмування, якою вона представлена</p> <p><i>використовує</i>: набуті знання для виконання поставлених вчителем завдань</p> <p><i>дотримується правил</i>: виконання тестових завдань</p> <p><i>показує на комп'ютері</i>: уміння щодо виконання тестових завдань</p>
<p><b>Тема 4: «Основи об'єктно-орієнтованого програмування. Робота у середовищі програмування», 6 годин</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>основні особливості середовища розробки, візуальне конструювання, структура проекту; структура модуля;</p>	<p><i>називає</i>: елементи візуального конструювання проектів, складові середовища програмування, етапи розробки проекту, етапи проектування форми</p> <p><i>наводить</i>: приклади етапів розробки проекту, етапів проектування форми</p> <p><i>розпізнає</i>: приклади програм, реалізованих у середовищі</p>

події; збереження та запуск проекту; компіляція; етапи розробки проекту; етапи проектування форми; проект як сукупність алгоритмів процедур обробки подій; кодування; налагодження; тестування

### **Практичні роботи:**

«Створення, збереження, запуск проекту за заданим зразком»

«Налагодження та тестування проекту за заданим зразком»

розробки, компонентів для введення та виведення інформації

*характеризує*: особливості використання мови об'єктно-орієнтованого програмування

*описує*: загальний вигляд середовища програмування, структури проекту

*пояснює*: правила оформлення проекту, алгоритм проектування форми

*формулює*: етапи розробки проекту, етапи проектування форми, поняття про об'єкт, властивості та події

*обґрунтовує*: доцільність використання середовища програмування для розробки проектів

*порівнює*: проекти, розроблені за допомогою середовища об'єктно-орієнтованого програмування та без його використання

*класифікує*: етапи розробки проекту у середовищі об'єктно-орієнтованого програмування

*аналізує*: переваги та недоліки використання середовища

об'єктно-орієнтованого програмування

**оцінює:** результати роботи реалізованого у вигляді проекту розробленого алгоритму для власних тестів у середовищі програмування

**висловлює судження:** щодо доцільності та коректності використання середовища програмування для конкретних задач

**розв'язує:** завдання по реалізації та тестуванню підготовлених вчителем алгоритмів з використанням операторів присвоювання, компонентів введення і виведення інформації, арифметичних виразів у середовищі програмування

**спостерігає:** за покроковим виконанням реалізованих у вигляді проекту запропонованих вчителем лінійних алгоритмів, за реалізацією проекту у середовищі програмування.

**використовує:** можливості середовища програмування для налагодження проектів, запропонованих вчителем,

	<p>при введенні початкової та при виведенні результуючої інформації</p> <p><i>дотримується правил:</i> роботи за комп'ютером, реалізації алгоритмів у вигляді проекту з використанням середовища програмування</p> <p><i>показує на комп'ютері:</i> виконання створеного проекту на основі запропонованих вчителем завдань, компоненти для введення початкової інформації, виведення результуючої інформації, виконання проекту для власних розроблених вхідних даних у середовищі програмування</p>
<p><b>Тема 5: «Лінійні алгоритми. Введення та виведення даних», 12 годин (10 годин)</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>поняття змінної; ім'я та тип змінної; опис стандартних типів змінних; арифметичні операції; правила запису арифметичних виразів; стандартні функції; оператор присвоювання; можливості введення та виведення інформації; використання текстових файлів для</p>	<p><i>називає:</i> арифметичні операції, стандартні функції, оператори присвоювання, можливості введення та виведення інформації, текстові файли як один із способів введення та виведення інформації</p> <p><i>наводить:</i> перелік стандартних типів змінних, приклади лінійних алгоритмів, арифметичних виразів, оператора присвоювання, можливостей введення і виведення інформації, текстових файлів для введення та виведення</p>

<p>введення та виведення інформації; поповнення власної бібліотеки навчальних завдань</p> <p><b>Практичні роботи:</b> «Побудова лінійних алгоритмів та їх реалізація у вигляді програм (проектів)», «Побудова лінійних алгоритмів та їх реалізація у вигляді програм (проектів) з використанням текстових файлів»</p> <p><b>Лабораторні роботи:</b> «Розробка власних лінійних алгоритмів та їх реалізація і тестування у середовищі програмування» «Розробка власних лінійних алгоритмів та їх реалізація і тестування у середовищі програмування з використанням текстових файлів»</p>	<p>інформації</p> <p><i>розпізнає</i>: відмінність між іменем та типом змінної, різні стандартні типи змінних, арифметичні операції, арифметичні вирази, стандартні функції, оператор присвоювання, способи введення та виведення інформації, текстові файли для введення та виведення інформації</p> <p><i>характеризує</i>: особливості різних стандартних типів змінних, арифметичні операції, правила запису арифметичних виразів, призначення оператора присвоювання, призначення способів введення та виведення інформації, текстові файли як один із способів введення та виведення інформації</p> <p><i>описує</i>: стандартні типи змінних, загальний вигляд оператора присвоювання, можливості введення та виведення інформації, виклик стандартних функцій, процес використання текстових файлів для введення та виведення інформації</p>
--	--

**пояснює:** поняття змінної, відмінність між іменем та типом змінної, призначення різних стандартних типів змінних, правила запису арифметичних виразів мовою програмування, оператора присвоювання, можливості введення і виведення інформації, схеми лінійного алгоритму, принцип виконання оператора присвоювання, принципи використання текстових файлів для введення та виведення інформації, результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих тестів

**формулює:** означення змінної, імені та типу змінної, означення арифметичного виразу, лінійного алгоритму

**обґрунтовує:** необхідність використання різних типів змінних для запису алгоритму у вигляді програми, пріоритетність виконання арифметичних операцій, особливості виклику стандартних функцій, переваги використання текстових файлів для введення та виведення інформації, значення тестування реалізованих

у вигляді програми (проекту) розроблених лінійних алгоритмів

**порівнює:** типи змінних, різні варіанти використання оператора присвоювання, введення та виведення інформації, способи введення та виведення інформації, результати виконання реалізованого у вигляді програми (проекту) розробленого лінійного алгоритму для запропонованих вхідних даних

**класифікує:** арифметичні операції, стандартні підпрограми, файли з вхідною та вихідною інформацією

**аналізує:** коректне визначення типів змінних щодо ефективного використання пам'яті комп'ютера при розв'язуванні конкретної задачі, відмінність між аргументами, результатами та проміжними величинами для заданих прикладів алгоритмів, результати роботи реалізованого у вигляді програми (проекту) розробленого лінійного алгоритму

**оцінює:** результати роботи реалізованого у вигляді

програми (проекту) розробленого лінійного алгоритму для власних вхідних даних

**висловлює судження:** щодо смислу введення різних типів змінних, необхідності використання коментарів при розробці алгоритмів у їх тексті, необхідності використання коментарів при введенні початкової інформації та при виведенні результуючої інформації, необхідності тестування розробленого алгоритму, необхідності використання можливостей середовища програмування під час налагодження лінійної програми (проекту)

**розв'язує:** завдання по створенню, реалізації мовою програмування та тестуванню лінійних алгоритмів з використанням операторів присвоювання, арифметичних виразів, стандартних функцій, різних способів введення і виведення інформації

**спостерігає:** за покроковим виконанням реалізованих у вигляді програм (проектів) розроблених лінійних

	<p>алгоритмів, за виконанням програм (проектів)</p> <p><b>використовує:</b> можливості середовища програмування для створення та налагодження програм (проектів), що базуються на лінійних алгоритмах, коментарі у тексті програми, при введенні початкової інформації та при виведенні результуючої інформації</p> <p><b>дотримується правил:</b> роботи за комп'ютером, реалізації лінійних алгоритмів у вигляді програм (проектів)</p> <p><b>показує на комп'ютері:</b> покрокове виконання створеної лінійної програми (проекту), введення початкової інформації, виведення результуючої інформації, можливості перегляду значень аргументів, результатів та проміжних величин, виконання лінійної програми (проекту) для власних розроблених вхідних даних</p>
<p><b>Тема 6: «Робота у середовищі програмування у консольному режимі», 4 години (*)</b></p> <p><b>Зміст навчального матеріалу теми:</b></p>	<p><b>називає:</b> консольний режим як одну із можливостей реалізації розробленого алгоритму у середовищі програмування; опції середовища програмування,</p>

поняття консольного режиму виконання програми та можливості його використання; знайомство з можливостями середовища об'єктно-орієнтованого програмування щодо роботи у консольному режимі; можливості створення, редагування та тестування програм у консольному режимі; можливості введення та виведення інформації для виконання програм у консольному режимі; використання текстових файлів для введення та виведення інформації; розв'язування задач у консольному режимі

**Практичні роботи:**

«Створення, редагування та тестування програм у консольному режимі»

«Використання текстових файлів для розв'язування задач у консольному режимі»

**Лабораторна робота:**

необхідні для виконання програми у консольному режимі  
*наводить*: приклади доцільності використання консольного режиму для виконання програми  
*розрізняє*: опції середовища програмування, які використовуються для виконання програми у консольному режимі  
*характеризує*: особливості використання консольного режиму для реалізації і виконання програм  
*описує*: етапи виконання програми у консольному режимі  
*пояснює*: можливості та послідовність етапів виконання програми у консольному режимі  
*формулює*: принципи роботи під час створення, редагування та тестування програми у консольному режимі  
*обґрунтовує*: доцільність використання консольного режиму для реалізації і виконання програм  
*порівнює*: процес створення, редагування та тестування програми з використанням форм та у консольному

«Реалізація лінійних алгоритмів з використанням консольного режиму середовища програмування»

режимі

**аналізує:** переваги та недоліки використання консольного режиму для створення, редагування та тестування програм

**оцінює:** необхідність використання консольного режиму для створення, редагування та тестування конкретної програми

**висловлює судження:** щодо особливостей, необхідності та доцільності використання консольного режиму для створення, редагування та тестування програм

**спостерігає:** за процесом налагодження програм у консольному режимі

**використовує:** можливості середовища програмування для створення, редагування та тестування програм

**дотримується правил:** роботи за комп'ютером, роботи у середовищі програмування у консольному режимі

**показує на комп'ютері:** покрокове виконання створеної лінійної програми, введення початкової інформації,

	<p>виведення результуючої інформації, виконання лінійної програми для власних розроблених вхідних даних у консольному режимі</p>
<p><b>Тема 7: «Алгоритми з розгалуженнями», 16 годин (12 годин)</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>логічні вирази та логічні операції; таблиці істинності; оператор розгалуження; вкладені розгалуження; оператор вибору</p> <p><b>Практичні роботи:</b></p> <p>«Побудова алгоритмів з послідовними розгалуженнями»,</p> <p>«Побудова алгоритмів з вкладеними розгалуженнями та оператором вибору»</p> <p><b>Лабораторні роботи:</b></p> <p>«Реалізація алгоритмів з послідовними розгалуженнями</p>	<p><i>називає:</i> логічні операції, два види логічних виразів, два види оператора розгалуження, оператор варіанта</p> <p><i>наводить:</i> приклади простих та складених логічних виразів, приклади виконання логічних операцій, приклади простих та вкладених розгалужень, приклади алгоритмів з використанням команди вибору</p> <p><i>розпізнає:</i> прості та складені логічні вирази, логічні операції <b>AND, OR, NOT</b>, скорочену і повну форму операторів розгалуження, послідовні та вкладені розгалуження</p> <p><i>характеризує:</i> особливості простих і складених логічних виразів, особливості виконання логічних операцій, особливості скороченої і повної форм розгалуження, послідовних і вкладених конструкцій розгалуження, особливості двох форм оператора вибору</p>

у вигляді програм (проектів)»,  
«Реалізація алгоритмів з вкладеними розгалуженнями та оператором вибору у вигляді програм (проектів)»

*описує*: загальний вигляд оператора умовного переходу у повній та скороченій формах, загальний вигляд оператора вибору у повній та скороченій формах

*пояснює*: відмінність між простими та складеними логічними виразами, відмінність між виконанням логічних операцій **AND**, **OR**, **NOT**, відмінність між скороченою і повною формою операції розгалуження, особливості використання послідовних і вкладених розгалужень

*формулює*: означення логічного виразу, простого логічного виразу, складеного логічного виразу, означення розгалуженого алгоритму

*обґрунтовує*: необхідність використання і застосування логічних виразів при складанні алгоритмів, необхідність існування двох форм розгалуження, необхідність коректного використання двох форм розгалуження, послідовних та вкладених розгалужень, оператора вибору, доцільність і значення тестування розроблених

алгоритмів з розгалуженнями

**порівнює:** дві різні форми розгалуження, послідовні та вкладені конструкції розгалужень, результати виконання реалізованого у вигляді програми (проекту) розробленого розгалуженого алгоритму для запропонованих вхідних даних

**класифікує:** розгалужені алгоритми за двома формами операторів розгалуження, за різними конструкціями розгалужень

**аналізує:** відмінність між простими та складеними логічними виразами, скороченою та повною формами розгалужень для заданих прикладів алгоритмів, результати роботи реалізованого у вигляді програми (проекту) розробленого розгалуженого алгоритму

**оцінює:** результати роботи реалізованого у вигляді програми (проекту) розробленого розгалуженого алгоритму для власних вхідних даних

**висловлює судження:** щодо коректності використання

простих або складених логічних виразів, скороченої або повної форми розгалуження, послідовних або вкладених розгалужень у заданих прикладах, необхідності використання можливостей середовища програмування під час налагодження розгалуженої програми (проекту)

**розв'язує:** завдання по створенню та тестуванню розгалужених алгоритмів з використанням операторів умовного переходу, вибору, різних розгалужених конструкцій

**спостерігає:** за покроковим виконанням реалізованих у вигляді програми (проекту) розроблених розгалужених алгоритмів, за виконанням програми (проекту)

**використовує:** можливості середовища програмування для створення та налагодження розгалужених програм (проектів), коментарі у тексті програми (проекту), при введенні початкової інформації та при виведенні результуючої інформації

**дотримується правил:** роботи за комп'ютером,

	<p>реалізації розгалужених алгоритмів у вигляді програм (проектів)</p> <p><i>показує на комп'ютері</i>: покрокове виконання створеної розгалуженої програми (проекту), введення початкової інформації, виведення результуючої інформації, виконання розгалуженої програми (проекту) для власних розроблених вхідних даних</p>
<p><i>II семестр (57 год. (54+3 рез.) ,3 год. на тиждень ) (38 год. (36+2 рез.), 2 год. на тиждень )</i></p>	
<p><b>Тема 8: «Алгоритми з повтореннями», 14 годин (10 годин)</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>алгоритми з повтореннями; оператори циклу мовою програмування; поєднання повторення і розгалуження; рекурентні послідовності*; вкладені цикли; можливості середовища програмування для роботи з циклічними програмами (проектами)</p> <p><b>Практичні роботи:</b></p>	<p><i>називає</i>: всі різновиди операторів повторення, можливості середовища програмування для роботи з циклічними програмами (проектами)</p> <p><i>наводить</i>: приклади повторень з передумовою, з післяумовою, з параметром; приклади виконання повторень з передумовою, з післяумовою, з параметром; рекурентних алгоритмів; вкладених циклів</p> <p><i>розпізнає</i>: циклічні алгоритми з передумовою, з післяумовою, з параметром; рекурентні алгоритми*, вкладені цикли</p>

«Побудова алгоритмів з послідовними повтореннями»  
«Побудова рекурентних алгоритмів»\*  
«Побудова алгоритмів з вкладеними повтореннями»

**Лабораторні роботи:**

«Реалізація алгоритмів з послідовними повтореннями у вигляді програм (проектів)»  
«Реалізація рекурентних алгоритмів у вигляді програм (проектів)»\*,  
«Реалізація алгоритмів з вкладеними повтореннями у вигляді програм (проектів)»

*характеризує:* особливості запису і виконання циклічних алгоритмів з передумовою, з післяумовою, з параметром; особливості створення і виконання рекурентних алгоритмів\*; особливості запису і виконання вкладених циклів

*описує:* загальний вигляд операторів повторення з передумовою, з післяумовою, з параметром; схематичне виконання рекурентних алгоритмів\*

*пояснює:* відмінність між виконанням повторень з передумовою, з післяумовою, з параметром; виконання рекурентних алгоритмів\* та вкладених циклів

*формулює:* принципи роботи повторень з передумовою, з післяумовою, з параметром; принципи роботи рекурентних алгоритмів\*, принципи роботи вкладених циклів

*обґрунтовує:* необхідність та доцільність використання і застосування повторень з передумовою, з післяумовою, з параметром; необхідність використання і застосування

рекурентних алгоритмів\* і вкладених циклів;  
необхідність і доцільність використання можливостей середовища програмування для налагодження циклічних програм (проектів); доцільність і значення тестування розроблених алгоритмів з повтореннями

**порівнює:** всі види повторень, послідовні та вкладені повторення, повторення з розгалуженнями та розгалуження з повтореннями

**класифікує:** алгоритми з повтореннями за трьома видами команд повторень

**аналізує:** відмінність між повтореннями з передумовою, з післяумовою, з параметром для заданих прикладів алгоритмів; відмінність між повтореннями з розгалуженнями та розгалуженнями з повтореннями, з вкладеними циклами, рекурентністю для заданих прикладів алгоритмів\*, результати роботи реалізованого у вигляді програми (проекту) розробленого циклічного алгоритму

**оцінює:** результати роботи реалізованого у вигляді програми (проекту) розробленого циклічного алгоритму для власних вхідних даних

**висловлює судження:** щодо коректності використання у заданих прикладах повторень з передумовою, з післяумовою, з параметром, рекурентних послідовностей\*, вкладених циклів, повторень з розгалуженнями, розгалужень з повтореннями; необхідності використання можливостей середовища програмування під час налагодження циклічної програми (проекту)

**розв'язує:** завдання по створенню та тестуванню циклічних алгоритмів з використанням всіх видів операторів повторення, різних циклічних конструкцій

**спостерігає:** за покроковим виконанням реалізованих у вигляді програми (проекту) розроблених циклічних алгоритмів, за виконанням програми (проекту)

**використовує:** можливості середовища програмування

	<p>для створення та налагодження циклічних програм (проектів), коментарі у тексті програми (проекту), при введенні початкової інформації та при виведенні результуючої інформації</p> <p><b>дотримується правил:</b> роботи за комп'ютером, реалізації циклічних алгоритмів у вигляді програм (проектів)</p> <p><b>показує на комп'ютері:</b> покрокове виконання створеної циклічної програми (проекту), введення початкової інформації, виведення результуючої інформації, виконання циклічної програми (проекту) для власних розроблених вхідних даних</p>
<p><b>Тема 9: «Масиви. Рядкові величини», 14 годин</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>поняття масиву; прості та структуровані типи мовою програмування; одновимірні та двовимірні масиви; рядкові величини; класичні алгоритми для роботи з масивами та рядками (пошук заданого елемента в</p>	<p><b>називає:</b> одновимірні та двовимірні масиви, цілочислові та дійсні масиви, символні масиви, процедури та функції для роботи з символними масивами, класичні алгоритми для роботи з масивами та рядками</p> <p><b>наводить:</b> приклади одновимірних та двовимірних масивів, символних масивів</p>

масиві, пошук та заміна підрядка, пошук мінімального або максимального елемента масиву, упорядкування масиву)

### **Практичні роботи:**

«Побудова алгоритмів з одновимірними і двовимірними масивами»

«Побудова алгоритмів з обробкою рядкових величин»

«Класичні алгоритми для роботи з масивами»

### **Лабораторні роботи:**

«Реалізація алгоритмів з одновимірними і двовимірними масивами у вигляді програм (проектів)»

«Реалізація алгоритмів з обробкою рядкових величин у вигляді програм (проектів)»

«Реалізація класичних алгоритмів для роботи з масивами та рядковими величинами у вигляді програм (проектів)»

**розпізнає:** алгоритми з використанням одновимірних та двовимірних масивів, символьних масивів

**характеризує:** особливості опису і використання одновимірних та двовимірних масивів, символьних масивів, особливості застосування класичних алгоритмів для роботи з масивами та рядками

**описує:** одновимірні та двовимірні масиви, символьні масиви, результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих вхідних даних

**пояснює:** принципи введення, виведення та обробки елементів масивів та рядкових величин

**формулює:** означення одновимірних, двовимірних масивів, індексу елемента масиву

**обґрунтовує:** доцільність використання масивів при розробці алгоритмів, необхідність застосування класичних алгоритмів для роботи з масивами та рядками для розв'язування алгоритмічних задач, значення

тестування реалізованих у вигляді програм (проектів)  
розроблених алгоритмів

**порівнює:** алгоритми без використання масивів та з їх застосуванням, результати виконання реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих вхідних даних

**класифікує:** алгоритми за розмірністю використаних масивів, за кількістю використаних масивів, за кількістю циклів, використаних в алгоритмі для введення, обробки та виведення елементів масивів, за класичними алгоритмами для роботи з масивами та рядками, зокрема алгоритмами упорядкування елементів одновимірних масивів

**аналізує:** відмінність між одновимірними та двовимірними масивами, між алгоритмами упорядкування елементів одновимірних масивів, особливості роботи з цілочисловими, дійсними, символьними масивами

**оцінює:** результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму з використанням масивів для власних вхідних даних

**висловлює судження:** щодо коректності використання у заданих прикладах одновимірних або двовимірних масивів, символічних масивів, класичних алгоритмів для роботи з масивами та рядками

**розв'язує:** завдання по створенню, реалізації у вигляді програм (проектів) та тестуванню алгоритмів з використанням масивів

**спостерігає:** за покроковим виконанням реалізованих у вигляді програм (проектів) розроблених алгоритмів з використанням масивів, за виконанням програм (проектів)

**використовує:** можливості середовища програмування для створення та налагодження програм з використанням масивів, класичні алгоритми для роботи з масивами та рядками при розв'язуванні алгоритмічних задач

	<p><i>дотримується правил</i>: роботи за комп'ютером, реалізації алгоритмів з використанням масивів у вигляді програм</p> <p><i>показує на комп'ютері</i>: покрокове виконання створеної програми (проекту) з використанням масивів, введення початкової інформації, виведення результуючої інформації, виконання циклічної програми (проекту) для власних розроблених вхідних даних</p>
<p><b>Тема 10: «Підпрограми. Процедури та функції», 14 годин</b></p> <p><b>Тема 10.1: «Процедури та функції», 10 годин</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>поняття підпрограми; локальні та глобальні змінні; формальні та фактичні параметри; створення підпрограм користувача та їх реалізація мовою програмування</p> <p><b>Практичні роботи:</b></p> <p>«Побудова алгоритмів з використанням процедур і</p>	<p><i>називає</i>: локальні та глобальні змінні, процедури і функції у мові програмування, формальні та фактичні параметри</p> <p><i>наводить</i>: приклад локальних та глобальних змінних, процедур і функцій у мові програмування, формальних та фактичних параметрів</p> <p><i>розпізнає</i>: локальні та глобальні змінні, процедури і функції, формальні та фактичні параметри</p> <p><i>характеризує</i>: особливості використання локальних та глобальних змінних, процедур і функції у мові</p>

функцій»

**Лабораторні роботи:**

«Реалізація алгоритмів з використанням процедур і функцій у вигляді програм (проектів)»

програмування, формальних та фактичних параметрів

*описує*: локальні та глобальні змінні, процедури і функції у мові програмування, формальне та фактичні параметри, організацію

*пояснює*: відмінність між локальними та глобальними змінними, формальними та фактичними параметрами, між процедурами та функціями, принципи виклику та виконання функцій та процедур, результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих вхідних даних

*формулює*: означення локальних та глобальних змінних, формальних та фактичних параметрів, загальний вигляд опису функцій та процедур

*обґрунтовує*: доцільність використання локальних та глобальних змінних, формальних та фактичних параметрів, функцій або процедур, значення тестування реалізованих у вигляді програми (проекту) розроблених алгоритмів

**порівнює:** принципи виклику та виконання функцій і процедур, результати виконання реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих вхідних даних

**класифікує:** алгоритми з використанням функцій і процедур

**аналізує:** відмінність між локальними та глобальними змінними, формальними та фактичними параметрами, організацією та виконанням функцій і процедур

**оцінює:** результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму з використанням процедур і функцій для власних вхідних даних

**висловлює судження:** щодо коректності використання у заданих прикладах процедур і функцій

**розв'язує:** завдання по створенню та тестуванню алгоритмів з використанням процедур і функцій

**спостерігає:** за покроковим виконанням реалізованих у

	<p>вигляді програми (проекту) розроблених алгоритмів з використанням процедур і функцій</p> <p><b>використовує:</b> можливості середовища програмування для створення та налагодження програм (проектів) з використанням процедур та функцій</p> <p><b>дотримується правил:</b> роботи за комп'ютером, реалізації алгоритмів з використанням процедур і функцій у вигляді програм (проектів)</p> <p><b>показує на комп'ютері:</b> покрокове виконання створеної програми (проекту) з використанням процедур і функцій, введення початкової інформації, виведення результуючої інформації, виконання програми (проекту) з використанням процедур і функцій для власних розроблених вхідних даних</p>
<p><b>Тема 10.2: «Рекурсія», 4 години (*)</b></p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>рекурсія; рекурсивні алгоритми; рекурсія у мові програмування</p>	<p><b>називає:</b> локальні та глобальні змінні, процедури і функції у мові програмування, формальні та фактичні параметри</p> <p><b>наводить:</b> приклади рекурсивних функцій та процедур</p>

**Практичні роботи:**

«Побудова рекурсивних алгоритмів»

**Лабораторні роботи:**

«Реалізація рекурсивних алгоритмів у вигляді програм (проектів)»

*розпізнає*: рекурсивні функції та процедури

*характеризує*: особливості рекурсивних функцій та процедур

*описує*: організацію рекурсивних функцій та процедур

*пояснює*: принципи роботи рекурсивних функцій та процедур, результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих вхідних даних

*формулює*: означення локальних та глобальних змінних, формальних та фактичних параметрів, загальний вигляд опису функцій та процедур

*обґрунтовує*: доцільність використання рекурсивних функцій або процедур, значення тестування реалізованих у вигляді програми (проекту) розроблених алгоритмів

*порівнює*: принципи виклику та виконання рекурсивних функцій і процедур, результати виконання реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих вхідних даних

**класифікує:** алгоритми з використанням рекурсивних функцій і процедур

**аналізує:** організацією та виконанням рекурсивних функцій і процедур

**оцінює:** результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму з використанням рекурсивних процедур і функцій для власних вхідних даних

**висловлює судження:** щодо коректності використання у заданих прикладах рекурсивних процедур і функцій

**розв'язує:** завдання по створенню та тестуванню алгоритмів з використанням рекурсивних процедур і функцій

**спостерігає:** за покроковим виконанням реалізованих у вигляді програми (проекту) розроблених алгоритмів з використанням рекурсивних процедур і функцій

**використовує:** можливості середовища програмування для створення та налагодження програм (проектів) з

	<p>використанням рекурсивних процедур та функцій</p> <p><b>дотримується правил:</b> роботи за комп'ютером, реалізації алгоритмів з використанням рекурсивних процедур і функцій у вигляді програм (проектів)</p> <p><b>показує на комп'ютері:</b> покрокове виконання створеної програми (проекту) з використанням рекурсивних процедур і функцій, введення початкової інформації, виведення результуючої інформації, виконання програми (проекту) з використанням рекурсивних процедур і функцій для власних розроблених вхідних даних</p>
<p><b>Тема 11: «Записи, множини, покажчики», 12 годин</b></p> <p>(*)</p> <p><b>Зміст навчального матеріалу теми:</b></p> <p>записи; оператор приєднання WITH; множини; покажчики; використання динамічної пам'яті</p> <p><b>Практичні роботи:</b></p> <p>«Побудова алгоритмів з використанням записів»</p>	<p><b>називає:</b> тип змінної «запис», оператор приєднання WITH, тип змінної «множина», тип змінної «покажчик»</p> <p><b>наводить:</b> приклади типів змінних «записи», оператора приєднання WITH, типів змінних «множини», типів змінних «покажчики»</p> <p><b>розпізнає:</b> змінні, що описані як записи, множини, покажчики</p> <p><b>характеризує:</b> особливості опису типів змінних «запис»,</p>

«Побудова алгоритмів з використанням множин»  
«Побудова алгоритмів з використанням покажчиків»

### **Лабораторні роботи:**

«Реалізація алгоритмів з використанням записів у вигляді програм (проектів)»  
«Реалізація алгоритмів з використанням множин у вигляді програм (проектів)»,  
«Реалізація алгоритмів з використанням покажчиків у вигляді програм (проектів)»

«множина», «покажчик», особливості використання оператора приєднання WITH

*описує:* типи змінних «запис», «множина», «покажчик», використання оператора приєднання WITH

*пояснює:* відмінність між використанням декількох масивів та типу «запис», відмінність між використанням масивів для збереження інформації та типу «множина», відмінність між використанням статичних та динамічних змінних, особливості використання типів змінних «запис», «множина», «покажчик», результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих вхідних даних

*формулює:* принципи застосування та специфіку розподілу пам'яті комп'ютера для змінних типів «запис», «множина», «покажчик»

*обґрунтовує:* доцільність і переваги використання типів змінних «запис», «множина», «покажчик», оператора приєднання WITH, значення тестування реалізованих у

вигляді програми (проекту) розроблених алгоритмів

**порівнює:** особливості використання записів та масивів, множин та масивів, статичної та динамічної пам'яті, результати виконання реалізованого у вигляді програми (проекту) розробленого алгоритму для запропонованих вхідних даних

**класифікує:** змінні за типами «запис», «множина», «показчик»; алгоритми за використанням статичної та динамічної пам'яті

**аналізує:** відмінність між використанням статичних масивів і змінних типу «запис», «множина», «показчик»

**оцінює:** результати роботи реалізованого у вигляді програми (проекту) розробленого алгоритму з використанням типів «запис», «множина», «показчик» для власних вхідних даних

**висловлює судження:** щодо доцільності та ефективності типів «запис», «множина», «показчик», динамічної пам'яті

***розв'язує:*** завдання по створенню та тестуванню алгоритмів з використанням типів «запис», «множина», «показчик», динамічної пам'яті

***спостерігає:*** за покроковим виконанням реалізованих у вигляді програм (проектів) розроблених алгоритмів з використанням типів «запис», «множина», «показчик», динамічної пам'яті

***використовує:*** можливості середовища програмування для створення та налагодження програм (проектів) з використанням типів «запис», «множина», «показчик», динамічної пам'яті

***дотримується правил:*** роботи за комп'ютером, реалізації алгоритмів з використанням типів «запис», «множина», «показчик», динамічної пам'яті

***показує на комп'ютері:*** покрокове виконання створеної програми (проекту) з використанням типів «запис», «множина», «показчик», динамічної пам'яті для власних розроблених вхідних даних

## ЛІТЕРАТУРА

1. Наказ Міністерства освіти і науки України № 834 від 27.08.10 року “Про затвердження Типових навчальних планів загальноосвітніх навчальних закладів III ступеню”.
2. Постанова Кабінету Міністрів України від 14.01.2004 № 24 “Про затвердження Державного стандарту базової і повної загальної середньої освіти”.
3. Книга вчителя інформатики: Довідково-методичне видання / Упоряд. Н.С.Прокопенко, Т.Г.Проценко – Харків: ТОРСІНГ ПЛЮС, 2005.– 256с.
4. Інструктивно – методичний лист про вивчення інформатики у 2010-2011 навчальному році.
5. Навчальна програма поглибленого вивчення інформатики для 8-12 класів (Т.П. Караванова, В.П. Костюков) // Інформатика. Програми для профільного навчання та допрофільної підготовки / К.: Видавнича група ВНУ, 2009. — стор. 278-368.
6. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.: Пер. с англ.: Уч. пос. — М.: Издательский дом “Вильямс”, 2000. — 384 с., ил.
7. Бондарев В.М., Рублинецкий В.И., Качко Е.Г. Основы программирования. Харьков: Фолио; Ростов н/Д: Феникс, 1997. — 368 с.
8. Вентцель Е.С. Исследование операций: задачи, принципы, методология. — 2-е изд., стер. — М.: Наука. Гл. ред. физ.-мат. лит., 1988. — 208 с. — (Пробл. науки и техн. прогресса)
9. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. — М.: Мир, 1989. — 360 с., ил.
10. Грузман М. Эвристика в информатике. — Винница: Арбат, 1998. — 308 с.

11. Караванова Т.П. Информатика: методи побудови алгоритмів та їх аналіз: необчислювальні алгоритми: .: Навч. посіб. для 9-10 кл. із поглибл. вивч. інф-ки – К.: Генеза. – 2006.- 216 с
12. Караванова Т.П. Информатика: основи алгоритмізації та програмування: 777 задач з рекомендаціями та прикладами: Навч. посіб. для 8-9 кл. із поглибл. вивч. інф-ки – К.: Генеза. – 2006.- 286 с.
13. Керман, Митчел, К. Программирование и отладка в Delphi. Учебный курс. Пер. с англ. — М.: Издательский дом “Вильямс”, 2002, 672 с.: ил. — Парал. тит.англ.
14. Кнут Д. Искусство программирования. — М.: Вильямс, 2000
15. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦНМО, 2001. — 960 с., 263 ил.
16. Культин Н.Б., Основы программирования в Delphi 2006 Microsoft .NET Framework-СПб. БХВ-Петербург, 2006 – 487с.:ил.
17. Липский В. Комбинаторика для программистов: Пер. с польск. — М.: Мир, 1988. — 213 с., ил.
18. Мандел Т. Разработка пользовательского интерфейса: Пер.с англ. — М. ДМК Пресс, 2001. – 416 с., ил. (Серия «Для программистов»)
19. Марков Е.П., Никифоров В.В. Delphi 2005 для .NET -СПб. БХВ-Петербург, 2005 – 896с.:ил.
20. Сван Т. Основы программирования в Delphi для Windows 95. – К.: «Диалектика», 1996. – 480 с., ил.
21. Стивенс Р. Тестирование и отладка программ на Visual Basic : Пер. с англ. — М.: ДМК Пресс, 2001. — 384 с. ил. (Серия «Для программистов»)
22. Тюкачев Николай, Свиридов Юрий Delphi 5. Создание мультимедийных приложений. —М: «Нолидж», 2000. –384 с., ил.
23. Шупруга В.В., Delphi 2006 на примерах СПб. БХВ-Петербург, 2006 – 528с.:ил.