ВІННИЦЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА КОЦЮБИНСЬКОГО

Факультет математики, фізики, комп'ютерних наук і технологій
Кафедра математики та інформатики

ДИПЛОМНА РОБОТА

на тему: **«Методичні особливості вивчення основ програмування мовою С++ в класах поглибленого вивчення інформатики»**

Студентки магістратури
Галузі знань 01 Освіта/ Педагогіка
Спеціальності014 Середня освіта (Інформатика)
Шелест Оксани Миколаївни

Використання чужих ідей, результатів і текстів мають посилання на відповідне джерело

_____ _____
(підпис) (ініціали, прізвище)

Науковий керівник
кандидат фізико-математичних наук, доцент Бак С. М.

Розширена шкала _____

Кількість балів: _____ Оцінка: ECTS_____

Голова комісії _____ _____
(підпис) (прізвище та ініціали)

Члени комісії _____ _____
(підпис) (прізвище та ініціали)

_____ _____
(підпис) (прізвище та ініціали)

_____ _____
(підпис) (прізвище та ініціали)

м. Вінниця – 2019

АНОТАЦІЯ

У роботі розглянуто особливості вивчення мови програмування С++ в класах поглибленого вивчення інформатики. Проаналізовані навчальні програми та підручники з інформатики. Розроблено тематичне і календарне планування навчального процесу. Також проаналізовано вибір методів, форм і засобів навчання та організацію оцінювання результатів навчання.

На основі проведеного дослідження запропоновано методику навчання основ програмування мовою С++ в класах поглибленого вивчення інформатики. Експериментально перевірено ефективність розробленої методики.

ANNOTATION

In the master's thesis, the features of studying the C++ programming language in the classes of in-depth study of computer science are considered. The curricula and textbooks on computer science are analyzed. Thematic and calendar planning of the educational process is developed. It also analyzes the choice of methods, forms and means of training and the organization of the assessment of learning outcomes.

Based on the study, a methodology for teaching the basics of programming in C ++ in the classes of in-depth study of computer science is proposed. The effectiveness of the developed methodology was experimentally verified.

ЗМІСТ

ВСТУП

Актуальність теми. Сьогодні технології розвиваються та поширюються по всьому світу швидкими темпами. Варто врахувати, що швидкість цього поширення буде тільки збільшуватися. Кожній сучасній дитині важливо розуміти, як і яким чином працюють програми та апаратне забезпечення. З розвитком інформаційних систем і технологій світ все більше потребує програмістів. Позиції, пов'язані з програмуванням, постійно знаходяться в топі рейтингу робіт.

Велика кількість мов програмування містить в собі принципи С і С++. Саме тому після їх вивчення в учнів не виникне труднощів у знайомстві з іншими мовами. Область використання також широка: створення мікроконтролерів, мобільних додатків, систем моделювання, ігор, нейронних мереж.

Питання навчання школярів основам програмування висвітлювали М.І. Жалдак, Н.В. Морзе, П.Г. Шевчук, В.В. Лапінський. Зокрема, П.Г.Шевчук описує методику навчання програмування учнів класів технологічного профілю та програмно-технологічні умови використання мови С++ для навчання програмування в загальноосвітніх навчальних закладах (Шевчук, 2011). В.В. Лапінськийна основі проведення експерименту серед учителів загальноосвітніх шкіл та учнів робить висновки, що головним у виборі мови програмування, якої спершу навчати школярів,має бути те,наскільки конкретний учитель готовий до викладання з використанням відповідного апаратно-програмного забезпечення (Лапінський, 2014).

Незважаючи на значну кількість науково-педагогічних і методичних праць щодо вивчення програмування у загальноосвітніх школах, проблема вивчення саме мови програмування С++ учнями середньої та старшої школи не достатньо розкрита на методичному рівні у вітчизняній науково-педагогічній літературі.

Мета дослідження – полягає в обґрунтуванні, розробці й експериментальній перевірці ефективності методики навчання програмуванню мовою С++ в класах поглибленого вивчення інформатики.

Об'єктом дослідження є процес навчання учнів програмуванню в закладах загальної середньої освіти.

Предметом дослідження є організаційно-методичні умови ефективного вивчення програмування мовою С++ в класах поглибленого вивчення інформатики.

Відповідно до об'єкта, предмета і мети визначено **завдання дослідження**:

1. З'ясувати науково-теоретичний і практичний стан проблеми дослідження.

2. Розробити методику навчання програмуванню мовою С++ в класах поглибленого вивчення інформатики.

3. Експериментально перевірити ефективність розробленої методики.

Практичне значення результатів дослідження. Результати дипломної роботи можуть бути використаними під час вивчення програмування в класах поглибленого вивчення інформатики.

Апробація результатів дипломної роботи. Результати дипломної роботи опубліковані у науково-популярному альманасі «Математика та інформатика навколо нас» ([37], [38]) і доповідались на науковому семінарі під час проходження науково-дослідної практики.

Структура роботи. Дипломна робота складається зі вступу, двох розділів (з висновками до них), висновку та списку використаної літератури (40 найменувань) та 3 додатків. Загальний обсяг – 73 сторінки.

РОЗДІЛ 1
ОСНОВИ АЛГОРИТМІЗАЦІЇ І ПРОГРАМУВАННЯ ЯК НЕОБХІДНА СКЛАДОВА ШКІЛЬНОГО КУРСУ ІНФОРМАТИКИ

У загальноосвітніх навчальних закладах предмет «Інформатика» з’явився більше 30 років тому.

Тоді з гасла "Програмування – це друге вміння читати і писати" було продумано зовсім інший підхід до викладання інформатики: "Найголовніше – це грамотний користувач, а програмування – для вузьких спеціалістів".

Синергія освіти та ІТ-галузі потребує більш ефективного розвитку держави в галузі інформаційних технологій при скоординованій діяльності вчителів та фахівців ІТ-галузі, ніж з однаковими зусиллями кожної людини.

Але якщо ІТ-індустрія зацікавлена у підборі кваліфікованих працівників, скажімо, що ІТ-дослідження в школах далекі від національних потреб ІТ-фахівців і призначені для підготовки переважно користувачів домашніх комп'ютерів або в ідеалі майбутніх професіоналів чи бухгалтерів, які знайомі з програмним забезпеченням Microsoft Office.

Хоча, прочитавши електронні документи різних державних відомств та установ, я розумію, що більшість сучасних операторів секретарів навіть не засвоїли мінімальні вимоги школи щодо обробки електронних документів.

Країна потребує як компетентних користувачів комп'ютерів, так і кваліфікованих фахівців з інформаційних технологій [34].

Не всі школи однаково оснащені комп’ютерними технологіями, оскільки немає фінансування. Не всі школи мають однаково добре розвинену групу учнів, а загальна сільська школа за рівнем учнів та вчителів все ще відрізняється від загальноосвітнього навчального закладу в столиці чи у великому місті.

1.1. Загальні відомості про мови програмування. Мова С++

Алгоритм — це послідовність точних команд виконавцеві алгоритму спрямованих на розв'язання певної задачі.

Виконавець алгоритму — це істота (людина, свійська тварина) або неістота (робот, автомат, комп'ютерна система), яка може виконувати всі вказівки заданого алгоритму.

Основні характеристики виконавця алгоритму:

- Середовище виконавця — умови, у яких може діяти виконавець.
- Елементарні дії — найпростіші дії, які може виконати виконавець.
- Система команд виконавця — сукупність допустимих команд виконавця.
- Допустимі команди — команди, які зрозумілі виконавцю і можуть бути ним виконані. Недопустимі команди — команди, які не можуть бути виконані виконавцем.
- Процес алгоритмізації — це визначення елементарних дій і порядку їх виконання та подання у вигляді послідовності команд виконавця.

Програма — це упорядкований список команд, написаних мовою програмування і призначених для виконання на комп'ютері. Отже, процес роботи комп'ютера полягає у виконанні програми, тобто набору цілком певних команд [31].

Мова програмування – це знакова система для опису алгоритмів програм, орієнтованих на конкретних виконавців (насамперед ЕОМ).

Класифікація мов програмування

Класифікацій мов програмування існує багато, але наукової теорії поки що немає. Три основні класифікації склалися історично:

1) за функціональною ознакою:

- універсальні мови (у них можна промоделювати, умовно кажучи, будь-який алгоритм);

- спеціалізовані мови (орієнтовані на певні класи задач);

2) за предметною орієнтацією:

Мови для розв’язання певного класу задач, наприклад, мови програмування для розв’язання задач символьної обробки (Lisp, Cobol), мови для обробки флеш-кліпів (ActionScript) тощо.

3) за рівнем абстракції:

- мови низького рівня (машинно-залежні) – Assembler тощо;
- мови високого рівня (орієнтовані на користувача до певної міри);
- Pascal, C, Fortran тощо;

Як окремий напрямок слід виділити мови програмування баз даних, призначені для маніпуляції великими централізованими масивами даних і отримання з них інформації. Багато з цих мов (Access, FoxPro, 4GL та ін.) мають розвинені процедурні елементи. Фактичним стандартом стала мова запитів до баз даних SQL [30].

Мови програмування низького рівня орієнтовані на конкретний тип процесора і враховують його особливості. Їх переваги: за допомогою мов низького рівня створюються ефективні і компактні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Недоліки: програміст, що працює з мовами низького рівня, має бути високої кваліфікації, добре розуміти будову комп’ютера, а також результуюча програма не може бути перенесена на комп’ютер з іншим типом процесора. Мови низького рівня, як правило, використовуються для написання невеликих системних додатків, драйверів пристроїв, модулів стиків з нестандартним обладнанням, коли найважливішими вимогами є компактність, швидкодія і можливість прямого доступу до апаратних ресурсів [2].

Мови програмування високого рівня дозволяють писати програми у формі, більш наближеній до звичайної мови. Програму, написану мовою високого рівня, можна більш легко читати і модифікувати, і це значно полегшує роботу програміста порівняно з написанням машинного коду. Для

перекладу програм, написаних мовою високого рівня, в машинні коди, повинні існувати спеціальні програми. Такі програми називаються трансляторами.

Мова програмування С++

С++ – це універсальна високорівнева мова програмування з підтримкою декількох парадигм програмування,зокрема, об'єктно-орієнтованого і процедурного. Розроблено Б'ярном Страуструпом в 1979 році і названо «С з класами». У 1990-х С++ стала найбільш широко використовуваною з мов програмування загального призначення.

Особливості мови програмування С++

При створенні С++ намагалися підтримувати сумісність з мовою С. Більшість програм на С будуть добре працювати з компілятором С++. С++ має синтаксис, заснований на синтаксисі С.

Нововведення С++ в порівнянні з С:

- підтримка об'єктно-орієнтованого програмування між класами;
- підтримка універсальних шаблонів програмування;
- доповнення до стандартної бібліотеці;
- додаткові типи даних;
- обробка винятків;
- простори імен;
- вбудовані функції;
- перевантаження оператора;
- перевантаження імені функції;
- посилання і оператори управління вільною пам'яттю.

Приклад першої та найпростішої програми "Hello, world!"

Приклад простої програми на С++, яка виводить Hello, world!" на стандартний вихідний канал.

```
#include <iostream>

 int main()
```

```
{
  std::cout << "Hello, world!" << std::endl;
  return 0;
}
```

Технічний огляд мови програмування C++

У 1998 році мова C++ була стандартизована Міжнародною організацією зі стандартизації під № 14882: 1998 – Мова програмування C++. Робоча група ISA в даний час працює над новою версією стандарту під кодовою назвою C++ 09 (раніше відомої як C++ 0X), яка повинна вийти в 2009 році.

Стандарт C++ на 1998 рік складається з двох основних частин: ядра мови і стандартної бібліотеки. Стандартна бібліотека C++ включає стандартну бібліотеку шаблонів STL. В даний час ім'я STL офіційно не використовується, але в колах програмістів на C++ це ім'я використовується для посилання на частину стандартної бібліотеки, яка містить визначення шаблонів, алгоритмів і функціоналів.

Стандарт C++ містить нормативну посилання на стандарт C з 1990 року і не визначає незалежно ті функції стандартної бібліотеки, які запозичені з стандартної бібліотеки C.

Однак існує величезна кількість нестандартних бібліотек C++. Багато бібліотек C можуть використовуватися в додатках C++.

Стандартизація визначила мову програмування C++, але неповні, обмежені, нестандартні версії мови можуть бути приховані під цим ім'ям. Спочатку мова розвивалася за формальними межами, спонтанно, перед обличчям завдань, які були перед ним поставлені. Розробка мови супроводжувала розробку крос-компілятора Cfront. Нововведення в мові відбито в зміні номера версії кросс-компілятора. Ці номери версій кросс-компілятора також поширювалися на самій мові, але поки немає мовних версій C++ [40].

Стандартна бібліотека

Стандартна бібліотека C++ включає стандартну бібліотеку C++ з незначними змінами, які роблять її більш важливою для C++. Решта бібліотеки C++ базується на стандартній бібліотеці шаблонів (STL). Він надає важливі інструменти, такі як контейнери (наприклад, вектори та списки) та ітератори (загальні покажчики), що дозволяють отримати доступ до цих контейнерів як масивів. Крім того, STL дозволяє працювати аналогічно іншим типам контейнерів, таким як асоціативні списки, стеки, черги.

Ви можете використовувати шаблони для написання загальних алгоритмів, які обробляють будь-який контейнер або послідовність, до членів яких звертаються ітератори.

Як і в C, функції бібліотеки активуються директивою #include для включення стандартних файлів. Стандарт C++ визначає 50 таких файлів.

STL до того, як ввійшов у C++, розроблявся сторонніми розробниками, спочатку HP, а потім SGI. Стандарт мови не називає його "STL", оскільки ця бібліотека стала невід'ємною частиною мови, але багато людей все ще використовують це ім'я, щоб відрізнити його від решти стандартної бібліотеки (Iostream, C тощо). Проект STLport, заснований на SGI STL, постійно оновлює STL, IOstream та класи рядків. Деякі інші проекти також стосуються розробки приватних стандартних бібліотечних додатків для різних завдань дизайну. Будь-який виробник компілятора C++, безумовно, забезпечить деяку реалізацію цієї бібліотеки, оскільки вона є дуже важливою частиною стандарту і широко використовується [41].

Нові функції порівняно з C

C++– це переважно підмножина C. Нові функції C++ включають вирази, перетворення типів, типи функцій, нові та видалені заяви, тип bool, посилання, розширена стійкість, функції за замовчуванням, аргументи за замовчуванням, простори імен, класи (включаючи всі пов'язані з функціональними класами, такі як успадкування, функції членів (методи),

віртуальні функції, абстрактні класи та конструктори), скасування операторів, шаблони, оператор ::, обробка винятків, динамічна ідентифікація тощо. C++ також у багатьох випадках суворо пов'язаний з контролем типу порівняно з C.

У C++ у попередника BCPL з'явився подвійний проріз ("//").

Пізніше деякі функції C++ були перенесені на C, такі як постійні та вбудовані ключові слова, циклічні оголошення та коментарі у стилі C++ ("//"). Пізніші C-впровадження також запровадили функції, недоступні в C++, такі як макроси vararg та поліпшена обробка поля параметрів.

Об'єктно-орієнтовані особливості мови

C++ додає об'єктно-орієнтовані можливості до C. Це стосується класів, які пропонують три важливі характеристики ООП: інкапсуляція, успадкування та поліморфізм.

У C структурах є основним способом організації даних. Ця структура містить ряд полів, які жодним чином не захищені. Якщо елементи структури мають змінну довжину, вони представляються як підказки. Виділення та розподіл пам’яті під цією увагою здійснюється вручну. Наприклад, розмірний масив змінної довжини на мові C з контролем кордону може бути представлений наступним чином:

```
struct Array {
   double* val;
   int len;
};

void FreeArray (const struct Array*);
void AllocArray (const struct Array*, int len);
double Elem (const struct Array*, int i);
void ChangeElem (const struct Array*, int i, double x);
```

Причини неефективності даної реалізації:

- Вам слід викликати FreeArray та AllocArray. Програміст може забути замінити одну з цих функцій або викликати її рано / пізно або двічі або із зазначенням недійсного масиву. Усі ці результати призводять до помилок, які важко виявити.
- Завдання Elem і ChangeElem повільні.
- Немає можливості замінити програмне забезпечення на інші функції для роботи зі структурою масиву. Ці функції можуть робити що завгодно із низькими та низькими полями.
- Немає можливості змінити програму безпосередньо в полях len і val;
- Відсутність набору структур призводить до руйнування домену до місця пам'яті. Ми не можемо це зупинити чи змінити.
- C++, використовуючи ООП, усуває всі ці проблеми.

Інкапсуляція

Основний спосіб організації інформації в C++– це класи. На відміну від типу структури (мови) мов C, які містять лише поля, клас C++ має поля членів та функції чи правила. У C++ код типу відрізняється від категорії до категорії, різниця полягає в тому, що при дотичних полях та функціях члени мають специфічну структуру та клас.

Маючи публічний простір, ви можете робити все поза аудиторією. Захищені та приватні зони поза класом не можуть бути захоплені, щоб запобігти введенню класу. Така спроба звернення призведе до помилки компіляції. Зовнішні члени захищені, а їхні поля не захищені. Навіть для читання. Цей вид захисту називається скасуванням [39].

Використовуючи blockchain, автор класу може захистити свої дані від помилок. Крім того, він розроблений для полегшення розвитку аудиторії. Результатом є те, що зміна видалення середовища проживання, незалежно від того, декларується вона публічно чи приватно, не потребує пов'язаних змін у класі, який використовує модифікований клас. Наприклад, якщо стара клініка зберігається під загальним іменем, а нова версія зберігається у вигляді

дерева, то сценарії, написані до зміни формату, потрібно записати, якщо дані є особистими або захищеними (у заключній ситуації – якщо Використовувані полюси – це не одне покоління, оскільки ці полюси не здатні запускати безпосередньо дані, але лише стандартні функції працюють належним чином у новій роботі в новій версії. Оператор доступу може бути встановлений як стандартна функція.

Функції учасників, а також домени можуть бути державними, захищеними та приватними. Можна зателефонувати в громадських справах, і лише члени та друзі можуть захищати їх та спілкуватися з ними.

Переваги та недоліки мови програмування С++

Переваги С++:

- Швидкість виходу програми С++ нижча, ніж програма С, хоча програмісти мають нові можливості та інструменти.
- масштабованість С++ розробляє програми для різних платформ і систем.
- Можливість роботи на низькому рівні з пам’яттю в порту (що при обережному використанні можна легко змінити за несприятливих умов)
- Можливість створення загальних алгоритмів для різних типів даних, які компетентні та обчислюють їх у процесі збору за допомогою шаблонів.

Недоліки С++:

Наявність безлічі функцій, що порушують принципи безпеки, призводить до того, що програми С++ легко можуть потрапити в серйозні помилки. Замість того, щоб контролювати компілятор, розробники змушені дотримуватися не дратівливих правил шифрування. По суті, ці правила обмежують С++ більш захищеним підкадром. Більшість проблем безпеки С++ успадковується від С, але автори мови відмовляються від автоматичного управління пам’яттю. (Як і збирання сміття) відіграє важливу роль у цьому

питанні. Тому візитні картки C++ стають уразливими для переповнення буфера.

Погана підтримка модулів. Підключення зовнішнього інтерфейсу модуля через вставлення заголовка препроцесора (#include) робить компіляцію дуже повільною, коли підключено багато модулів. Для вирішення цього питання багато компіляторів використовують заздалегідь складений механізм заголовка. Відсутність типу збору даних (CTTI).

Удосконалений процесор C++ (успадкований від C) – дуже примітивна система. Це призводить до того, що неможливо (або важко) виконувати певні завдання з програмування, з іншого боку, через їх первісний характер вони часто призводять до помилок і вимагають великих дій. Щоб уникнути проблем, які можуть виникнути Деякі мови програмування (наприклад, Scheme і Nemerle) мають системи мета програмування. Це більш потужне і безпечне (також відоме як макрос [3].

З кінця XX століття так звані програми мета програмування на основі шаблонів стали всюдисущими у спільноті C++. По суті, вони використовують функції шаблонів C++ для використання мов програмування, традиційної роботи, яка виконується під час компіляції відповідно до них. Сама по собі ця особливість дуже цікава. Але з вищезазначених причин попередній код дуже важко розпізнати та налагодити. Lisp / Scheme, Nemerle та інші ефективніші та водночас простіші для розуміння підсистеми. Крім того, D можна порівняти з енергією. Але дуже проста у використанні в підсистемі шаблонів

Хоча було встановлено, що C++ є багатопарадигматичною мовою, але вона не сумісна для кодування для роботи мовами. Деякі з цих упущень були усунені кількома бібліотеками (Loki, Boost), які використовують інструменти. метапрограмування Розширення мови шляхом створення функції (наприклад, вона підтримує лямбд / анонімний метод), але якість цього рішення не поступається. Робоча мова вбудована у рішення. Мови, що

використовуються, такі як зразкове відображення, важко емулювати метапрограмами.

Критика C++

C++ успадкував ряд питань мови C:

Завдання проекту позначається як "=", а функція порівняння "==". Вони легко взаємозамінні, і така структура буде належним чином синхронізована, але це серйозна помилка. Це особливо часто зустрічається з командами, якщо, наприклад, програміст може записати if (i = 0) замість if (i == 0) (але більшість перекладачів надсилає попередження в таких випадках). якщо записати всі функції порівняння так: якщо (0 == i). Крім того, багато мов (BASIC, Pascal) використовують символ " = =" для порівняльних операцій.

Призначте (=), Збільшити (++), Зменшити (-) та інші повернені значення. У поєднанні з багатьма функціями це дозволяє розробнику, але не вимагає, щоб код був важким для читання. Навпаки, один з основних принципів C і C++– це дозволяти розробникам писати в будь-якому стилі і не встановлювати "хороший" стиль. Крім того, це іноді дозволяє компілятору генерувати більш економічний код [33].

Макроси (#define) – потужні, але небезпечні пристрої. У C++, на відміну від C, потреба в небезпечному множенні набагато рідше через інтегрованих шаблонів та функцій. Але у старих стандартних бібліотеках C багато небезпечних макросів.

Деякі вважають, що дискомфорт у C ++ є недоліком системи поводження з відходами. З іншого боку, у C++ є багато інструментів (категорії конструкторів та руйнівників, стандартні шаблони, змінні посилання), які дозволяють практично виключити використання небезпечних покажчиків. Однак відсутність інтегрованого збору відходів дозволяє користувачеві обирати свою політику управління ресурсами.

Крім того, значно зменшується збір відходів, що є руйнівним, оскільки важлива ефективність.

Розвиток мови програмування C++

C++ постійно розвивається для задоволення сучасних вимог. Boost – одна з нинішніх мовних груп C++, яка надає вказівки Комітету зі стандартизації C++ щодо шляхів його вдосконалення. Наприклад, одним із цих атрибутів у цій групі є вдосконалення мовних навичок шляхом додавання функцій синхронізації.

Стандарт C++ не описує методи іменування об'єктів, деякі винятки для обробки деталей та інші функції, пов'язані з виконанням, що робить код, сформований деякими перекладачами, несумісним. З цією метою треті сторони запровадили ряд стандартів для конкретних архітектур та операційних систем [8].

Але станом на цей текст, він прагне до повного впровадження стандарту C++ серед перекладачів C++, особливо в області шаблонів, що є частиною мови, нещодавно розробленої Комітетом зі стандартизації.

Однією з перешкод для цієї проблеми є експорт ключових слів, які також використовуються для розбиття реклами та визначення шаблону.

Comeau C++ став першим компілятором, який підтримав експорт шаблону на початку 2003 року (через п'ять років після виходу стандарту). У 2004 році він також почав підтримувати бета-версії Borland C++ Builder X.

Обидва компілятори базуються на зовнішньому інтерфейсі EDG. Інші перекладачі, такі як Microsoft Visual C++ або GCC, взагалі не підтримують це. Через серйозні проблеми з повним впровадженням, Герб Саттер, секретар Комітету зі стандартизації C++, рекомендував виключити експорт майбутніх версій стандарту, але потім вирішив відмовитися від нього.

1.2. Місце і роль основ алгоритмізації і програмування у шкільному курсі інформатики

Інформатика як освітня дисципліна швидко розвивається. Якщо раніше базовий курс інформатики складався з вивчення основ алгоритмізації та програмування, основ пристрою і застосування обчислювальної техніки, то сьогодні метою курсу інформатики в школі є підвищення ефективності застосування людиною комп'ютера як інструмента. Комп'ютерна грамотність визначається не тільки умінням програмувати, а, в основному, умінням використовувати готові програмні продукти, розраховані на призначений для користувача рівень. Розробка програмно-інформаційних засобів є вельми дорогою справою через його високу наукоємність та необхідності спільної роботи висококваліфікованих фахівців: психологів, комп'ютерних дизайнерів, програмістів. Проте вона окупає себе завдяки тому, що доступ до комп'ютера сьогодні може одержати практично кожна людина навіть без спеціальної підготовки.

Основи алгоритмізації і програмування є одним із найбільш складних розділів при вивченні курсу інформатики. В даний час існує велика кількість проблем у даному напрямку. Однією із проблем є проблема вибору вчителем мови програмування. Сьогодні дедалі частіше в класах поглибленого вивчення інформатики (або в класах інформаційно-технологічного профілю) вчителі обирають мову С++, яка стала універсальною мовою для програмістів усього світу [7].

При побудові навчання учнів темі «Мова програмування С++» кожен вчитель інформатики стикнеться з величезною кількістю питань: як побудувати виклад матеріалу, які використовувати методичні розробки, в якій формі проводити заняття, які скласти практичні завдання,який матеріал використовувати учням при вивченні. Всі ці питання виникають через відсутність чітко і в повному обсязі викладених навчально-методичних матеріалів для вивчення даної теми.

Кожен шкільний підручник з інформатики включає в себе різні розділи, пов'язані з вивченням інформаційно-комунікаційних технологій та основ інформатики. Але в жодному підручнику не розглядається тема «Мова програмування С++». Тому завдання вчителя в школі буде полягати в розробці такої методики, яка максимально спростить розвиток здатності програмувати, що дуже важливо для більшості людей в сучасному технічному світі. Вчителю доведеться користуватися особистими розробками уроків, використовувати раніше напрацьований досвід і підручники, які прямо чи опосередковано містять матеріал для вивчення обраної мови програмування [23].

Можна запропонувати побудувати навчання за темою «Мова програмування С++» послідовно таким чином, щоб учні на початковій стадії навчання ознайомилися з різними мовами програмування, змогли зрозуміти необхідність вивчення даної мови програмування. На початку вивчення учні повинні ознайомитися з основними алгоритмічними конструкціями, командами мови, правилами опису об'єктів мови програмування, структурою програми і правилами написання. Учням необхідно спочатку сформувати навички написання найпростіших програм з використанням алгоритмічних конструкцій та основних об'єктів мови програмування, а потім перейти до вивчення простих і далі більш складних методів програмування.

Вивчення мови програмування в школах необхідне для формування знань, умінь і навичок програмування, а також для формування абстрактного, логічного і алгоритмічного мислення в учнів.

1.2.1. Аналіз навчальних програм з інформатики

Обговорення шкільної програми з інформатики для 5-9 класів, показує, що у вчителів і викладачів немає єдиної думки щодо її змісту. Тому вважаємо доцільним проаналізувати існуючі навчальні програми з інформатики для учнів 5-9-х класів для того, щоб виявити, які теми вивчаються з метою навчання учнів середніх класів програмуванню.

Навчальна програма для учнів 10-11 класів загальноосвітніх навчальних закладів (Рівень стандарту)

Програма розрахована на вивчення інформатики в 10-11 класах закладів загальної середньої освіти усіх профілів суспільного-гуманітарного, філологічного, художньо-естетичного та спортивного напрямів, а також фізичного, біолого-хімічного, біолого-фізичного, біолого-географічного, біотехнологічного, хіміко-технологічного, фізико-хімічного та агрономічного профілів природничо-математичного напряму на рівні стандарту обсягом 1 година на тиждень. Тобто це та програма по якій навчається переважна більшість шкіл в нашій країні і саме цю програму ми маємо враховувати в першу чергу при вивченні теми «Мова програмування С++». Провівши аналіз програми треба перш за все відмітити, що тем, пов’язаних з вивченням мов програмування не багато. У пояснювальній записці вказуються такі завдання:

- виховання основ інформаційної культури учнів;
- формування в учнів бази знань, умінь і навичок, необхідних для кваліфікованого та ефективного використання сучасних інформаційно-комунікаційних технологій у навчально-пізнавальній діяльності та повсякденному житті, а також для самостійного опанування новими інформаційними технологіями, що з’являтимуться у майбутньому.

У другому пункті вказується, що база знань, уміння і навички які сформовані в учнів мають відповідати рівню кваліфікованого користувача

сучасного ПК. Саме в цьому і є протиріччя у цій програмі: між тими завданнями які ставить на меті дана програма і як вони виконуються [25].

Навчальна програма для учнів 10-11 класів загальноосвітніх навчальних закладів (Академічний рівень).

Програма розрахована на вивчення інформатики в 10–11 класах загальноосвітніх навчальних закладів фізико-математичного та екологічного профілю природничо-математичного напряму в обсязі 1 година на тиждень у 10 класах та 2 години на тиждень у 11 класі.

Проаналізувавши цю програму можна сказати, що вона відрізняється від програми рівня стандарту лише тим що у 11 класі вивчення предмету інформатики збільшилося на 1 годину. Таким чином суттєвого впливу вивчення мов програмування не має. Отже так само в програмі академічного рівня також вивчення теми, як і в базовому рівні пропонується для розгляду на факультативних заняттях.

Навчальна програма поглибленого вивчення інформатики для учнів 8-11 класів загальноосвітніх навчальних закладів

Програма курсу розрахована на вивчення інформатики за варіантом постійного використання комп’ютерів. Вивчення курсу сплановано на 4 роки (8-11 кл.) з розрахунку 3 години на тиждень у 8-9 класах, 4 години на тиждень у 10-11 класах). Але і в цій програмі пропонують вивчати програмумання в середовищі Lazarus або інтерпретовану об'єктно-орієнтовану мову програмування Python [26].

На вивчення теми "Основи алгоритмізації програмування" відводиться різна кількість годин.Порівняльний аналіз навчальних програм з інформатики для 5-9 класів наведений в таблиці 1.1.

Таблиця 1.1

Порівняльний аналіз навчальних програм з інформатики для 5-9 класів

№	Назва програми	Клас	Теми з програмування	Години
1	Навчальна програма для 5-9 класів загальноосвітніх навчальних закладів для класів, які вивчали інформатику в 2-4 класах	5	Алгоритми та програми	12
		6		12
		7		12
		8		24
		9		18
	Навчальна програма для 5-9 класів загальноосвітніх навчальних закладів	5	Основи алгоритмізації та програмування	-
		6		7
		7		9
		8		28
		9		10
	Програма з інформатики для 5-9 класів загальноосвітніх навчальних закладів з поглибленим вивченням предметів природничо - математичного циклу	5	Алгоритмізація і програмування	7
		6		5
		7		5
		8		25
		9		25
	Навчальна програма для 8-9 класів загальноосвітніх навчальних закладів з поглибленим вивченням окремих предметів	8	Основи алгоритмізації та програмування	56
		9		56

Отже підводячи підсумок по аналізу нових програм з інформатики можна з упевненістю сказати, що увага не приділяється на вивчення мови програмування С++. Отже виходячи з цього необхідно розробити програму

для вивчення мови програмування на факультативних або гурткових заняттях з інформатики, що є одним із завдань магістерської роботи.

1.2.2. Аналіз підручників з інформатики

На сучасному етапі розвитку освіти суспільство ставить високі вимоги до рівня навченості, тому завдання сучасної школи полягає у вихованні особистості, здатної до творчої діяльності. На сьогоднішній день інформатика розглядається як один з найважливіших компонентів загальної освіти, що відіграє велику роль у розв’язанні пріоритетних завдань навчання та виховання – у формуванні цілісного світогляду, картини світу, навчальних та комунікативних навичок, основних психічних якостей учнів. Вивчення інформатики та інформаційних технологій у школі є невід'ємною частиною сучасної освіти. Завданням шкільного курсу «Інформатика» є здатність справляти різнобічний навчальний, розвивальний і виховний вплив на учнів, сприяти формуванню особистості, здатної правильно обрати свій шлях у житті, зважаючи на власні можливості, рівень компетентності та конкурентоспроможності.

Для вивчення інформатики можна використовувати такі навчально-методичні матеріали, підручники:

Інформатика (рівень стандарту). Підручник для 10 (11) класу закладів загальної середньої освіти. Автори: Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А., Шакотько В. В. Вивчаючи даний підручник учні 10-11 класів продовжать вивчати основи інформатики. Кожний розділ складається з пунктів, які, у свою чергу, містять підпункти. В даному підручнику відсутній розділ "Основи алгоритмізації та програмування" [29].

Інформатика (рівень стандарту). Підручник для 10 (11) класу закладів загальної середньої освіти. Автори: Морзе Н.В, Барна О.В.. Даний підручник реалізує базовий модуль навчання інформатики у профільній школі. Інформатика входить до переліку предметів за вибором, які надають учням можливість удосконалювати знання з обраної галузі шляхом вивчення поглиблених модулів з фаху або споріднених з ним. А це означає, що майбутнє учнів буде тісно пов’язане з інформаційними технологіями, подальший професійний вибір буде відповідати реаліям і запитам

інформаційного суспільства, а отримані знання та навички в курсі інформатики стануть міцним фундаментом формування цифрової компетентності, яка забезпечить успіх та реалізацію цілей учнів. Підручник складається із чотирьох розділів, що містять як теми, які узагальнюють та систематизують набуті знання та уміння із базового курсу інформатики 5–9 класів, так і ті, що розкривають нові напрями розвитку технологій, їх застосування для навчання та життя в умовах високотехнологічного суспільства [24].

У підручнику багато завдань і вправ, для розв’язування яких, потрібно застосовувати набуті знання та вміння в процесі навчання та навички роботи з прикладними програмами. В даному підручнику також відсутній розділ "Основи алгоритмізації та програмування".

Проаналізувавши програмне забезпечення з інформатики для школи, можна зробити висновок, що вільне програмне забезпечення є більш раціональним для використання, спираючись на існування трьох основних мотивів для використання вільного програмного забезпечення державними навчальними організаціями:

1. усунення залежності від одного постачальника програмного забезпечення;
2. зменшення витрат на інформатизацію;
3. безпека роботи за комп’ютером.

Основним підручником з інформатики для 10-11 класів на сьогодні є підручник «Інформатика» . Автори А.М. Гуржій, І.Т. Зарецька, Б.Г. Колодяжний, О.Ю. Соколов. Автори підручника вважають, що курс інформатики можна поділити на 5 частин:

1. Загальний курс користувача.
2. Поглиблений курс користувача.
3. Основи алгоритмізації та програмування
4. Інформаційне моделювання. Етапи розв’язування задач за допомогою комп’ютера.

5. Спеціалізація.

Перша частина підручника охоплює загальний курс користувача (загальні поняття інформатики, апаратне забезпечення комп'ютера, коротка характеристика основних видів програмного забезпечення, операційні системи (MS-DOS та Windows), графічний та текстовий редактори, основи роботи в глобальній мережі Інтернет), у другій частині висвітлено поглиблений курс користувача (ознайомлення з електронними таблицями і системами управління базами даних) та основи алгоритмізації та програмування.

Матеріал, викладений у підручнику, значно перевищує за обсягом той, який необхідний за програмою (зокрема, детально розглянуті основні положення теорії баз даних, алгоритми сортування і пошуку), і може бути використаний у школах, ліцеях, гімназіях, де курс починається раніше, наприклад, у 8 класі. При такому вивченні автори пропонують розподілити матеріал підручника таким чином: 8 клас — загальний курс користувача, 9–10 класи — поглиблений курс користувача, основи алгоритмізації та програмування, етапи розв'язування задач, 11 клас — спеціалізація.

Підручник орієнтований на вивчення курсу інформатики у школах, у яких комп'ютери працюють як під управлінням операційної системи MS-DOS, так і під управлінням Windows.

У підручнику наведена велика кількість завдань та вправ різної складності. Їх кількість авторами свідомо збільшена, що забезпечує гнучкість у підборі завдань залежно від контингенту учнів, дає можливість проведення додаткових і факультативних занять. Частину з них автори пропонують використати для тематичної атестації.

Матеріал навчального посібника «Інформатика. 10–11 клас». Автор Я.М. Глинський охоплює тематичні розділи курсу інформатики як із вивчення основ алгоритмізації та програмування, так і з вивчення інформаційних технологій. Як і попередній підручник, складається з двох

частин, проте відрізняється поділом матеріалу. Як перевагу посібника можна відзначити наявність інструкцій для практичних робіт.

Навчальний матеріал посібника «Базовий курс інформатики» . Автори В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу розрахований на вивчення інформатики протягом чотирьох років, але може застосовуватися для вивчення цієї дисципліни і протягом двох років та в інших формах і обсягах. У посібнику детально розглянуто як теоретичні основи базового курсу інформатики, так і методи використання популярного програмного забезпечення.

Посібник є одним з перших, де описано українізовані версії найсучасніших програмних продуктів корпорації Microsoft та інших виробників. Усі розділи є логічно завершеними самостійними модулями. Матеріал викладено доступною для учнів мовою, текст не обтяжений громіздкими означеннями й формулюваннями, супроводжується ілюстраціями та детальними поясненнями методів роботи з конкретними програмами.

Посібник складається з двох книг: у першій розглядаються базові теми курсу, вивчення яких достатньо для підготовки кваліфікованого користувача персонального комп’ютера, друга присвячена найважливішим сучасним інформаційним технологіям (програмування, системи керування базами даних, веб-дизайну).

Підручник не є єдиним засобом навчання. Для ефективного навчання повинен бути єдиний комплекс взаємопов’язаних між собою книг, наочних посібників, електронних засобів навчального призначення.

Найбільш ґрунтовною працею з методики навчання інформатики є навчальнийпосібник «Методика навчання інформатики» Н.В.Морзе (за редакцією М.І. Жалдака). Друга частина якого присвячена особливостям ознайомлення учнів з поняттям інформації, вивчення апаратного забезпечення, основних видів прикладного програмного забезпечення.

У додатках також наведено матеріали для факультативних занять. Ці матеріали можна використовувати і як додатковий матеріал при проведенні уроків.

Перелік методичних посібників збільшується. Крім того, широкий спектр методичних матеріалів міститься у журналах «Комп'ютер у школі та сім'ї», «Інформатика та інформаційні технології в навчальних закладах», газеті «Інформатика» тощо.

Проаналізувавши підручники які пропонуються для вивчення шкільного курсу інформатики можна зробити висновок, що в них відсутні розділи , які б допомагали учням у вивчені саме мови програмування С++.

ВИСНОВКИ ДО РОЗДІЛУ 1

Розвиток технологій і пов’язані з цим зміни в світі призводять до того, що програмування, яке вивчається переважно на позакласних заняттях штурмом вривається у сферу обов’язкових наук, щоб заповнити прогалину фахівців ІТ поміж випускниками шкіл, яких так потребують у вищих навчальних закладах та на ринку праці. У той же час з’являється все більше сигналів, що програмування має неоціненну роль у розвитку навичок учнів, а саме: логічне мислення, вирішення проблем, творчість і співпраця, тобто тих навичок, які є найбільш цінними. Раннє запровадження науки програмування у школах буде великим викликом для сучасної освіти. Проте, його не потрібно боятися так як проведені дослідження дозволили розробити відповідні інструменти, які дозволяють легко та у привабливий спосіб представити програмування з перших років шкільної освіти. Навчатися програмуванню за допомогою роботів та ігор дітям дуже цікаво, в той же час діти здобувають інструменти та навички, плоди використання яких можна буде збирати впродовж довгих років.

Проаналізувавши програму та перелік підручників, які запропоновано для вивчення інформатики,можна зробити висновок, що хоча навчальні програми строго не регламентують вибір мови програмування для вивчення, проте у шкільних підручниках не передбачено вивчення мови програмування С++. Разом з тим, вибір мови програмування для вивчення основ програмування залишається за вчителем інформатики, тому все більше вчителів обирають мову С++, як одну із найпопулярніших серед розробників, для вивчення основ програмування в класах поглибленого вивчення інформатики або в класах інформаційно-технологічного профілю. Однак сьогодні бракує методичних матеріалів, а тим більше шкільних підручників, для вивчення цієї мови. Недостатньо вивченим є питання методичних особливостей навчання програмуванню мовою С++ в класах поглибленого вивчення інформатики.

РОЗДІЛ 2
МЕТОДИКА НАВЧАННЯ ПРОГРАМУВАННЮ МОВОЮ С++ В КЛАСАХ ПОГЛИБЛЕНОГО ВИВЧЕННЯ ІНФОРМАТИКИ

2.1. Тематичне і поурочне планування навчального процесу

Нами розроблено тематичне планування з теми "Основи програмування мовою С++",розраховане для учнів, які навчаються в класах поглибленого вивчення інформатики (або інформаційно-технологічного профілю) закладів загальної середньої освіти.

Метою вивчення даної теми є: формування теоретичної бази знань учнів та практичних навичок розв'язування задач.

Даний курс розрахований на 70 годин (2 години на тиждень). Розподіл годин між темами "Основи програмування С++" подано в таблиці 2.1.

Таблиця 2.1

Розподіл годин між темами "Основи програмування С++"

Розподіл годин між темами "Основи програмування С++"				
№ п/п	Тема	Теоретичні години	Практичні години	Разом
1	Основні поняття мови програмування С++	3	0	3
2	Перша програма на С ++	3	2	5
3	Змінні і типи даних в С ++	1	3	4
4	Конструкція розгалуження в С ++	3	1	4
5	Цикли С++	3	3	6
6	Масив в С++	4	2	6
7	Функції в С ++	3	1	4
8	Покажчики в С ++	2	2	4
9	Динамічні масиви в С++	3	1	4

10	Параметри командного рядка в С ++	3	1	4
11	Класи в С ++	5	1	6
12	Вектори в С ++	2	2	4
13	Спадкування класів в С ++	2	2	4
14	Перевантаження функцій в С ++	3	1	4
15	Перевантаження методів класу в С ++	3	1	4
16	Визначення і перевантаження операторів класу в С ++	4	0	4
			Всього	70

Розроблене нами відповідне календарно-тематичне планування подано в таблиці 2.2.

Таблиця 2.2

Календарно-тематичне планування

№ уроку	Дата уроку	Тема уроку	Примітки
ТЕМА 1. ОСНОВНІ ПОНЯТТЯ МОВИ ПРОГРАМУВАННЯ С++ (4 год.)			
1.		Первинний інструктаж з БЖД. Безпека життєдіяльності при роботі з комп'ютером. Історія створення та загальна характеристика С++.	
2.		Інструктаж з БЖД. Алфавіт.	
3.		Інструктаж з БЖД. Ідентифікатори.	
ТЕМА 2. ПЕРША ПРОГРАМА НА С++ (4 год)			
4.		Інструктаж з БЖД. Перша програма мовою С++.	
5.		Інструктаж з БЖД. Практична робота 1. Знайомство з середовищем програмування.	
6.		Інструктаж з БЖД. Загальна структура програми.	

		Коментарі.	
7.		Інструктаж з БЖД. Практична робота 2. Базові навики роботи з середовищем С++.	
8.		Інструктаж з БЖД. Практична робота 3. Створення програми, що відображає повідомлення.	
ТЕМА 3. ЗМІННІ І ТИПИ ДАНИХ В С++ (4 год)			
9.		Інструктаж з БЖД. Ініціалізація та оголошення змінних.	
10.		Інструктаж з БЖД. Основні типи даних в С++.	
11.		Інструктаж з БЖД. Практична робота 4. Створення програми "простий калькулятор на С++".	
12.		Інструктаж з БЖД. Практична робота 5. Аналіз коду програми.	
ТЕМА 4. КОНСТРУКЦІЯ РОЗГАЛУЖЕННЯ В С++ (4 год)			
13.		Інструктаж з БЖД. Блок Оператор послідовного обчислення.	
14.		Інструктаж з БЖД. Оператор if та else .	
15.		Інструктаж з БЖД. Практична робота 6. Створення програми з використанням операторів if та else.	
16.		Інструктаж з БЖД. Оператор switch ,оператори циклу, оператори переходу.	
ТЕМА 5. ЦИКЛИ В С++ (6 год)			
17.		Інструктаж з БЖД. Поняття циклу.	
18.		Інструктаж з БЖД. Види циклів, Whilesum+=digit, Do while, Цикл for.	
19.		Інструктаж з БЖД. Практична робота 7. Створення циклічних програм з лічильником мовою С++.	
20.		Інструктаж з БЖД. Практична робота 8. Створення циклічних програм з умовою на мові С++.	
21.		Інструктаж з БЖД. Ключові слова. Оператори continue	

		та break.	
22.		Інструктаж з БЖД. Практична робота 9. Створення програм з циклами та розгалудженнями мовою С++.	
ТЕМА 6. МАСИВ В С++ (6 год)			
23.		Інструктаж з БЖД. Визначення та елементи масива.	
24.		Інструктаж з БЖД. Індекси масивів.	
25.		Інструктаж з БЖД. Практична робота 10 . Створення програм обробки одновимірних масивів величин мовою С++.	
26.		Інструктаж з БЖД. Оголошення масивів фіксованого розміру.	
27.		Інструктаж з БЖД. Динамічні масиви.	
28.		Інструктаж з БЖД. Практична робота 11 . Створення програм обробки двовимірних масивів величин мовою С++.	
ТЕМА 7. ФУНКЦІЇ В С++(4 год)			
29.		Інструктаж з БЖД. Оголошення та визначення функцій.	
30.		Інструктаж з БЖД. Типи функцій.	
31.		Інструктаж з БЖД. Способи передачі параметрів і повернення результату обчислень функцій	
32.		Інструктаж з БЖД. Практична робота 12 . Створення програм для опрацювання функцій мовою С++.	
ТЕМА 8. ПОКАЖЧИКИ В С++ (4 год)			
33.		Інструктаж з БЖД. Визначення покажчика.	
34.		Інструктаж з БЖД. Керовані та некеровані покажчики.	
35.		Інструктаж з БЖД. . Практична робота 13 . Створення програми з використанням керованих покажчиків.	
36.		Інструктаж з БЖД. . Практична робота 14 . Створення програми з використанням некерованих покажчиків.	
ТЕМА 9. ДИНАМІЧНІ МАСИВИ В С++ (4 год)			

37.		Інструктаж з БЖД. Оператор динамічного виділення пам’яті.	
38.		Інструктаж з БЖД. Оператор звільнення динамічної пам'яті .	
39.		Інструктаж з БЖД. Динамічні двовимірні масиви.	
40.		Інструктаж з БЖД. Практична робота 15 . Створення програми з використанням масивів.	
ТЕМА 10. ПАРАМЕТРИ КОМАНДНОГО РЯДКА В С++(4 год)			
41.		Інструктаж з БЖД. Інтерфейс командного рядка.	
42.		Інструктаж з БЖД. Параметри командного рядка.	
43.		Інструктаж з БЖД. Поточна директорія.	
44.		Інструктаж з БЖД. Практична робота 16. Створення директорії.	
ТЕМА 11. КЛАСИ В С++ (6 год)			
45.		Інструктаж з БЖД. Поняття класу і об’єкту в мові С++.	
46.		Інструктаж з БЖД. Використання класів.	
47.		Інструктаж з БЖД. Принципи побудови класу.	
48.		Інструктаж з БЖД. Відкриті і закриті члени класу.	
49.		Інструктаж з БЖД. Використання конструкторів та деструкторів.	
50.		Інструктаж з БЖД. Практична робота 17. Створення програм з використанням класів на мові С++.	
ТЕМА 12. ВЕКТОРИ В С++ (4 год)			
51.		Інструктаж з БЖД. Визначення вектора.	
52.		Інструктаж з БЖД. Управління елементами вектора .	
53.		Інструктаж з БЖД. Методи класу vector.	
54.		Інструктаж з БЖД. Практична робота 18. Створення вектора.	
ТЕМА 13. СПАДКУВАННЯ КЛАСІВ В С++ (4 год)			

55.		Інструктаж з БЖД. Похідні класи.	
56.		Інструктаж з БЖД. Конструктор базового класу.	
57.		Інструктаж з БЖД. Практична робота 19. Створення базового класу.	
58.		Інструктаж з БЖД. Практична робота 20. Спадкування від базового класу.	
ТЕМА 14. ПЕРЕВАНТАЖЕННЯ ФУНКЦІЙ В С++ (4 год)			
59.		Інструктаж з БЖД. Перевантаження функцій.	
60.		Інструктаж з БЖД. Типи повернення в перевантаженні функцій.	
61.		Інструктаж з БЖД. Псевдоніми типів в перевантаженні функцій.	
62.		Інструктаж з БЖД. Практична робота 21. Виклики функцій.	
ТЕМА 15. ПЕРЕВАНТАЖЕННЯ МЕТОДІВ КЛАСУ В С++ (4 год)			
63.		Інструктаж з БЖД. Суть перевантаження методу у класі.	
64.		Інструктаж з БЖД. Сигнатур методу.	
65.		Інструктаж з БЖД. Параметри з модифікаторами ref і out	
66.		Інструктаж з БЖД. Практична робота 22. Створення програм з використанням перевантажених методів.	
ТЕМА 16. ВИЗНАЧЕННЯ І ПЕРЕВАНТАЖЕННЯ ОПЕРАТОРІВ КЛАСУ С++ (4 год)			
67.		Інструктаж з БЖД. Операторні функції	
68.		Інструктаж з БЖД. Перевантаження унарних та бінарних операторів за допомогою функцій-членів.	
69.		Інструктаж з БЖД. Перевантаження бінарних операторів за допомогою дружніх функцій.	
70.		Інструктаж з БЖД. Перетворення типів.	

2.2. Організація навчально-пізнавальної діяльності учнів під час вивчення мови програмування С++

Оскільки саме при навчанні основ програмування закладається як теоретична, так і практична база підготовки майбутніх інженерів-програмістів, актуальною є проблема визначення особливостей методичної системи навчання основ програмування в школі, яка б сприяла активізації навчально-пізнавальної, дослідницької діяльності учнів розкриттю їх творчого потенціалу, розвитку самостійності та індивідуальних здібностей особистості й ґрунтувалася на широкому впровадженні у навчальний процес новітніх педагогічних та інформаційно-комунікаційних технологій і враховувала міжнародні стандарти щодо підготовки фахівців у галузі інформаційно-комунікаційних технологій і комп’ютерних наук.

На уроках інформатики комп'ютер є і предметом вивчення, і засобом навчально-пізнавальної діяльності, що відповідним чином впливає на організацію навчального процесу. Специфіка уроку інформатики виявляється, передусім, в істотному обсязі практичних робіт з використанням комп'ютера, при якому «контактний час» роботи з комп'ютером становить майже половину уроку [15].

На уроках ми пропонуємо використовувати різні види робіт, а саме групові форми роботи, індивідуальні роботи та роботи в парах. Для розвитку організаційно-діяльнісних якостей учнів застосовую різні способи утворення груп. Групи створюю на основі вже існуючого розміщення учнів у класі. Даний спосіб має формальну основу, але потребує найменших часових затрат. Склад учнівських груп визначаю жеребкуванням. Спосіб є ефективним. Також дозволяла учням самостійно об'єднується в групи по 4 - 6 осіб. Це найбільш природний само організуючий спосіб при умові наявності необхідного часу. Спочатку за

певними критеріями обираю лідерів майбутніх груп, які потім набирають собі в групи інших учнів.

Особливої уваги заслуговує технологія організації роботи в групах. Оскільки групи працюють в основному самостійно, їх необхідно цього навчати. Обов'язково на початку уроку проводився загальний інструктаж та заздалегідь готувала завдання.

При роботі в групах учні навчаються таких видів діяльності:

- підготовка виступу перед класом, демонстрація роботи програми,
- колективне обговорення розв'язування поставленої задачі методом «мозкового штурму»,
- підготовка учнів до взаємодії з іншими групами — підготовка для них питань, конкурсів і змагань, участь груп в розв’язуванні спільної для всього класу задачі,
- виконання творчого завдання — вивчення нової прикладної програми, розробка проекту.

У роботі груп переважають такі види діяльності учні ставлять цілі, планують свою роботу, обговорюють проблеми, що виникають, розподіляють роботу в груш, контролюють, аналізують і оцінюють свою діяльність, проводять рефлексію. Способи обговорення в групі можуть бути різними. Найбільше ефективно на першому етапі виявилося висловлювання своєї думки всім членам групи. Це дисциплінувало учнів, привчало стежити за своєю мовою, В кінці кожного уроку в групах підводився підсумок, що зроблено, як працювали, які завдання на домашнє опрацювання [11].

2.2.1. Вибір методів, форм і засобів навчання

На уроках інформатики ми пропонуємо використовувати такі методи навчання, які на нашу думку дозволяють ефективно побудувати навчальний процес з урахуванням специфічних особливостей учнів:

- пояснювально-ілюстративний;
- репродуктивний;
- бесіда;
- ігрові методи;
- проблемно-пошуковий;
- метод проектів.

Звичайно,це не весь перелік методів, а лише ті методи, які найчастіше використовуються на уроках. Кожен учитель повинен підбирати методи, які забезпечать сприймання матеріалу в повному обсязі. Використання пояснювально-ілюстративного методу під час вивчення теми "Основні поняття мови програмування С++" забезпечило правильне сприйняття навчального матеріалу, усвідомлення та запам'ятовування учнями історії виникнення, основних визначень та загальну характеристику мови програмування. Цінність цього методу полягає в тому, що він сприяє засвоєнню і відтворенню значного обсягу знань.

Використання репродуктивного методу під час вивчення тем "Знайомство з середовищем програмування", " Загальна структура програми. Коментарі", "Базові навики роботи з середовищем С++". Даний метод заснований на відтворені знань, повторені способів діяльності за завданням вчителя. І з цією метою використовується неодноразове виконання одного і того самого завдання, а також варіативних, схожих із раніше засвоєними зразками. Оскільки курс інформатики має чітку практичну спрямованість, то на етапі формування умінь та навичок учнів ніяк не обійтися без репродуктивного методу навчання. Учитель може пояснювати правила виконання операцій, проводити демонстрації, використовувати для цього

різні засоби навчання, але доки учень сам не виконає простого відтворення (репродукції) дії вчителя, у нього ніколи не сформуються відповідні навички.

Використання способу бесід під час вивчення теми "Змінні і типи даних у С++" . Особливість цього методу навчання полягає в тому, що учні відтворюють або сприймають інформацію частинами, у формі запитання – відповідь. За рівнем пізнавальної самостійності учнів у процесі навчання застосовується два види бесіди: евристична і репродуктивна.

Репродуктивна бесіда – це система репродуктивно-мнемонічних і репродуктивно-пізнавальних запитань. Вони спонукають до відтворення учнями засвоєних знань і оволодіння готовими знаннями з різних джерел: підручника, засобів наочності, спостережень, дослідів . Репродуктивну бесіду як метод навчання використовувала на усіх етапах уроку [1].

Еврестична бесіда – це спосіб організації творчої діяльності школярів через розв'язання проблеми у співпраці з учителем. Функція вчителя у цій бесіді полягає не тільки у постановці системи запитань, керуванні пошуком нових знань і способів діяльності, а й у показі способу отримання відповідей на ті запитання-підпроблеми, на які учні не можуть відшукати самостійно. Результатом еврестичної бесіди є нові знання та уміння.

Використання ігрового методу навчання під час вивчення теми "Цикли" та "Масиви" передбачає застосування у навчанні елементів ігрової діяльності, внаслідок чого дидактичне завдання стає більш зрозумілим, доступним і привабливим для дитини, а процес навчання цікавішим. Були проведені такі ігри: "Логіка програміста", "Допиши визначення, формулу, програму", "Зроби висновок", "Інформаційне лото", "Зайвий термін", а також розгадування кросвордів та ребусів.

Проблемно-пошуковий (еврестичний) метод навчання використовувався під час виконання практичних робіт. Даний метод призначений для активізації розумової діяльності учнів. Головне під час використання цього методу – правильно сформулювати проблему, а також стимулювати самостійний пошук учнями шляху її розв'язання.

Використання методу проектів допомагає реалізувати проблемне навчання як активізує і поглиблює пізнання, дозволяє навчати самостійному мисленню і діяльності, системному підходові в самоорганізації, дає можливість навчати груповій взаємодії.

Застосування даного методу дозволяє вирішувати одночасно цілий ряд задач:

- активізувати самостійну роботу учнів за допомогою проектно-дослідницької діяльності;
- познайомити учнів з принципами використання об'єктно - орієнтованого програмування;
- навчити учнів використовувати програмне забезпечення для представлення своїх досліджень;
- створити позитивну мотивацію у вивченні предмета;
- навчити учнів доводити свою точку зору, відповідаючи на питання по проекті;

Засоби навчання – це різноманітні матеріали і знаряддя навчального процесу, завдяки яким більш успішно і за коротший час досягається визначена ціль навчання. До засобів навчання належать: підручники, навчальні посібники, дидактичні матеріали, технічні засоби (ТЗН), обладнання, станки, навчальні кабінети, лабораторії, ЕОМ, ТБ та інші засоби масової комунікації. Засобами навчання можуть також бути реальні об'єкти, виробництво, споруди. Дидактичні засоби, як і методи, організаційні форми, є частиною педагогічної системи. Вони виконують такі основні функції: інформаційну, засвоєння нового матеріалу, контрольну. Вибір засобів навчання залежить від дидактичної концепцій мети, змісту, методів, форм і умов навчального процесу [4].

На уроках було застосовано як прості так і складні засоби, а саме словесні, візуальні, аудіальні та аудіовізуальні.

За допомогою інтернет - засобу LearningApps.org можна створювати інтерактивні вправи. Даний онлайн сервіс є конструктором для розробки різноманітних завдань для використанням і на уроках, і позаурочний час.

За допомогою онлайн сервісу учні можуть перевірити і закріпити свої знання в ігровій формі, що сприяє формуванню їх пізнавального інтересу. Сервіс LearningApps надає можливість отримання коду для того, щоб інтерактивні завдання були розміщені на сторінки сайтів або блогів вчителів і учнів.

Сайти для самостійного вивчення С++:https://ocw.mit.edu, https://www.edx.org, https://codeasy.net, https://www.coursera.org, https://install.convertmyvid.com, https://code-live.ru.

Інтернет ресурси: https://exercism.io, https://www.codewars.com.

Для організації навчальної діяльності учнів в експериментальній групі нами було розроблено сайт https://sites.google.com/view/shelestoksana-vdpu. На цьому сайті розміщено теоретичний матеріал, практичні завдання, тести.

Результати опитування учнів "як ви відноситесь до наявності сайта вчителя інформатики?" подано на діаграмі

Рис. 2.1 Результати опитування учнів про наявність сайту

2.2.2. Організація оцінювання результатів навчання

Оцінювання навчальних досягнень учнів з інформатики здійснюється відповідно до загальних критеріїв оцінювання.

При вивченні курсу інформатики передбачається проведення різних видів практичної діяльності учнів: тренувальних, практичних робіт, які є лише частиною уроку інформатики.

Основними функціями оцінювання навчальних досягнень учнів є:

- контролююча визначає рівень досягнень кожного учня (учениці), готовність до засвоєння нового матеріалу, що дає змогу вчителеві відповідно планувати й викладати навчальний матеріал;
- навчальна сприяє повторенню, уточненню й поглибленню знань, їх систематизації, вдосконаленню умінь та навичок;
- діагностико-коригувальна з'ясовує причини труднощів, які виникають в учня під час навчання; виявляє прогалини у засвоєному, вносить корективи, спрямовані на їх усунення;
- стимулювально-мотиваційна формує позитивні мотиви навчання;
- виховна – сприяє формуванню умінь відповідально й зосереджено працювати, застосовувати прийоми контролю й самоконтролю, рефлексії навчальної діяльності [5].

Тестова перевірка все більше набуває поширення. Сутність цього методу полягає в тому, що учням у певному дидактичному блоці визначають конкретні завдання (запитання), на які подані альтернативні відповіді. Учень має обрати правильну відповідь. Якщо йти таким спрощеним шляхом, то це може призводити до простого вгадування відповіді. Важливо моделювати завдання в такий спосіб, щоб учень аргументував свій вибір відповіді, аналізував, чому інші відповіді він вважає помилковими чи неповними. Тестова перевірка може здійснюватися машинним і безмашинним способом [9].

Нами розроблено серію тестів по темах: "Цикли С++", "Функції в С++", "Масиви в С++", "Класи в С++". Тести були створені за допомогою Google Forms. Переваги даних опитувань є: не потрібно тиражувати матеріал опитування на всіх респондентів; немає потреби особисто спілкуватися з респондентами, збирати їх всіх в один час і в одному місці; результати опитування зберігаються і постійно доступні в Інтернеті.

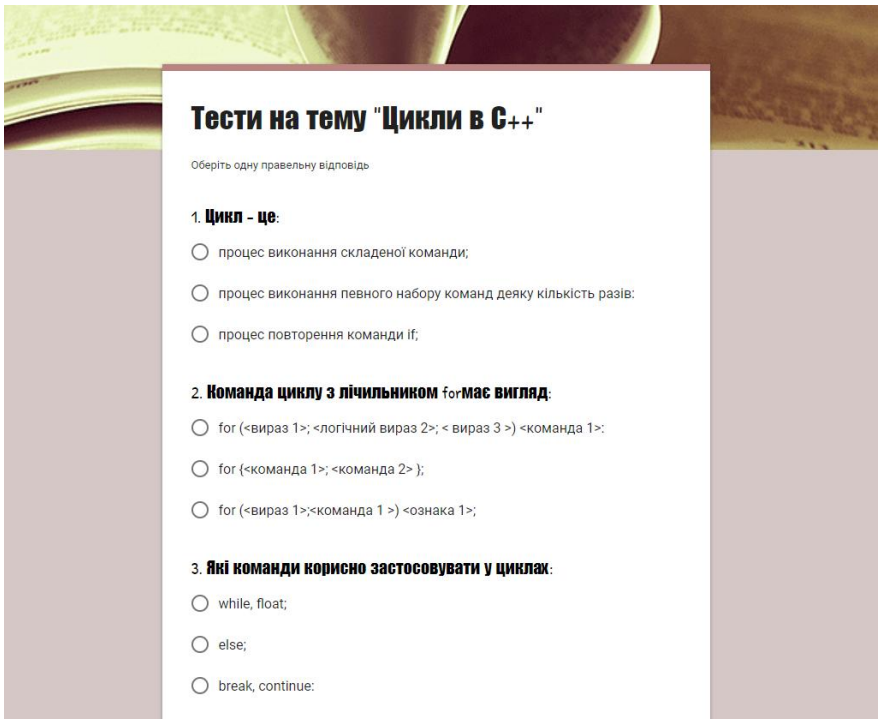

Рис. 2.2 Тест створений за допомогою Google Forms

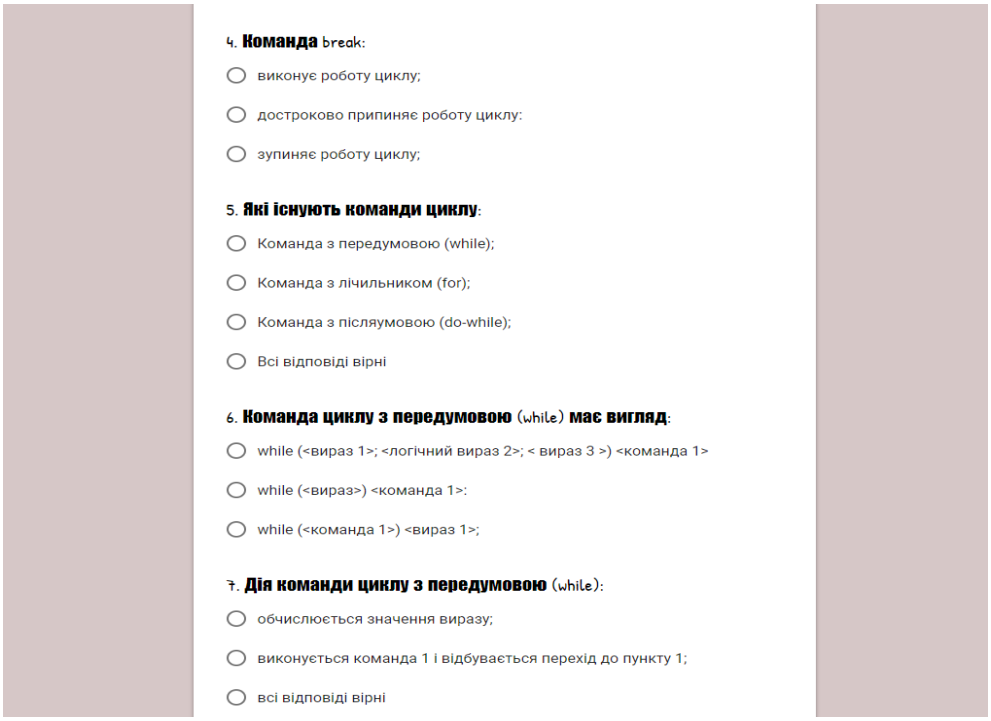

Рис. 2.3 Тест створений за допомогою Google Forms

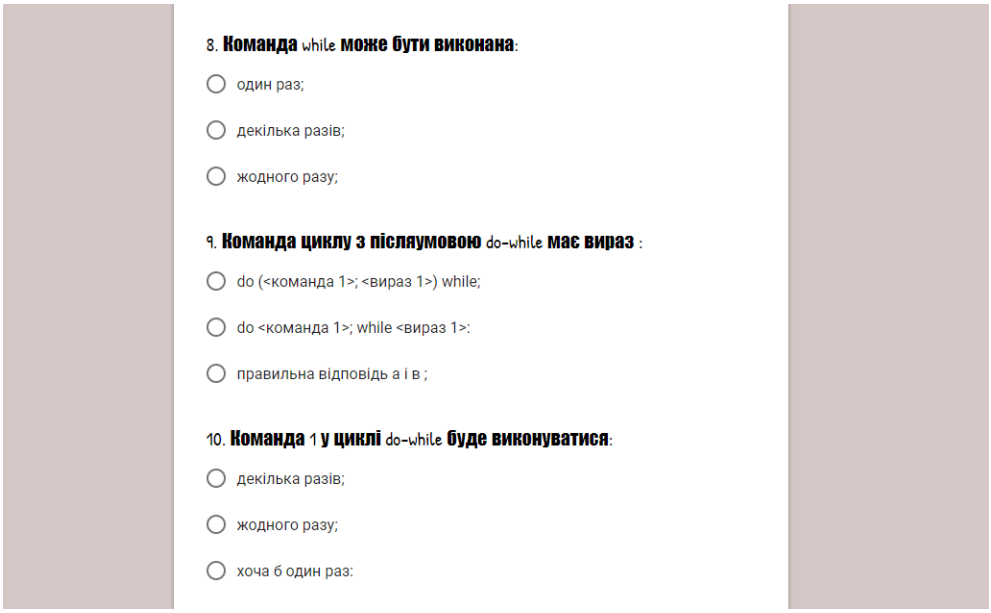

Рис. 2.4 Тест створений за допомогою Google Forms

2.3. Експериментальна перевірка ефективності розробленої методики та аналіз результатів дослідження

Експериментальною базою дослідження стала Вороновицька середня загальноосвітня школа I-III ступенів №1. Для експериментальної перевірки розробленої методики проводились факультативні заняття з інформатики за розробленою нами програмою. Перед початком експерименту серед учнів 8 – 11 класів було проведено опитування «Чи хочу відвідувати факультатив з програмування?». Результати опитування наведено в таблиці 2.3.

Таблиця 2.3

Чи хочу відвідувати факультатив з програмування?

Чи хочу відвідувати факультатив з програмування?			
Класи	Так	Ні	Невпевнений
8-А	9	11	8
8-Б	8	9	11
9-А	8	10	8
9-Б	9	12	7
10-А	11	5	6
10-Б	13	5	2
11-А	5	4	8
11-Б	4	3	7

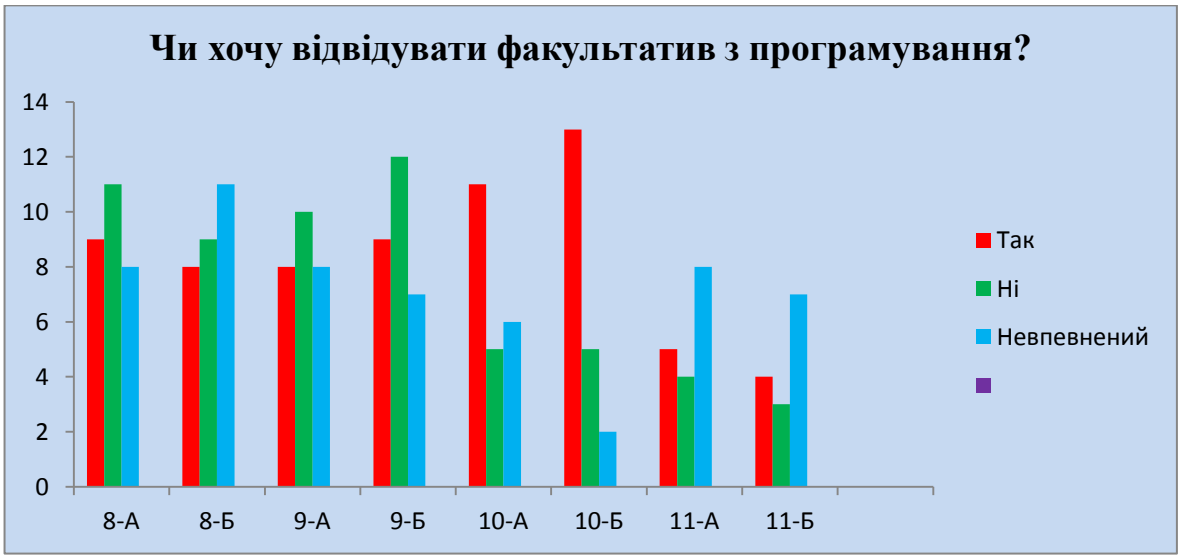

Рис. 2.5 Результати опитування учнів «Чи хочу відвідувати факультатив з програмування?»

Дослідженням було охоплено два 10-х класи, які проявили найбільший інтерес до факультативу з програмування (24 з 42 учнів). Зауважимо, щ о згадані класи є класами технічного профілю.

Серед 24 бажаючих відвідувати факультатив ми додатково провели опитування «Чи хочу я вивчати мову програмування С++?».

Таблиця 2.4

Чи хочу я вивчати мову програмування С++?

Чи хочу вивчати мову програмування С++			
	Так	Ні	Невпевнений
Учні, які записалися на факультатив	83,33 %	12,50 %	4,17%

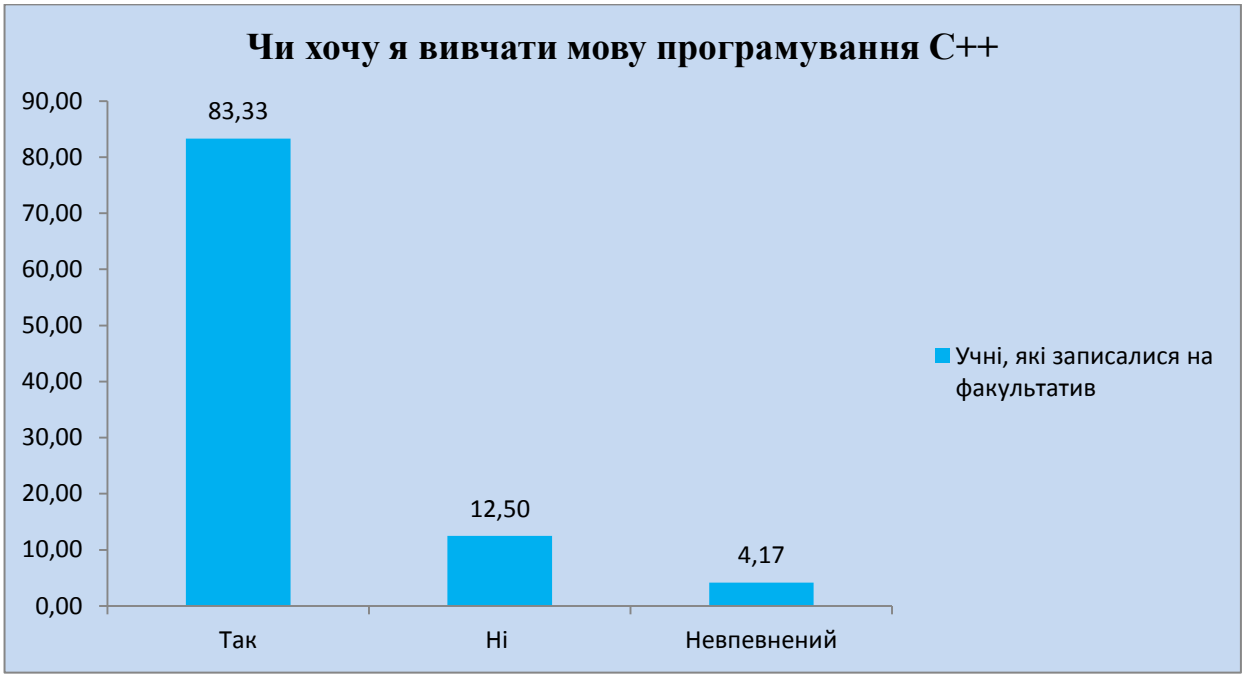

Рис. 2.6 Результати опитування учнів «Чи хочу вивчати мову програмування С++?»

Рис. 2.7 Самооцінка рівня мотивації до вивчення програмування мовою С++

Аналізуючи результат опитування, для проведення факультативних занять з програмування мовою С++ було обрано 20 учнів з високим і середнім рівнем мотивації вивчати С++.

З метою проведення експериментального дослідження набрану для факультативу групу було розділено на дві підгрупи по 10 учнів: контрольну та експериментальну. В першій застосовувались традиційні методи, а експериментальній – традиційні та інноваційні методи відповідно до розробленої методики.

У ході дослідження визначені основні методичні підходи залучення учнів у активну навчально-пізнавальну діяльність під час вивчення мови програмування С++:

- формування розуміння учнями ООП як цілісного поняття інформатики, що покладено в основу розробки і використання складних програмних систем;
- забезпечення оволодіння учнями теорією та набуття практики програмування;
- розкриття необхідності і корисності засвоєння нових знань.

Мета експериментального дослідження:

- виявити рівень знань учнів безпосередньо з програмування на початку і в кінці дослідження;
- проаналізувати методи, форми і засоби навчання, визначити, які з них найбільш ефективні для вивчення мови програмування С++. На основі цього розробити власну методику вивчення програмування мовою С++;
- перевірити ефективність розробленої методики.

Для досягнення мети експериментального дослідження були розв'язані такі завдання:

- Визначення показників для вимірювання рівня знань у програмуванні.
- Виділення рівнів розвитку учнів в процесі навчання програмуванню.
- Уточнення шляхів та методичних прийомів розвитку пізнавальної активності в процесі навчання мови програмування С++.
- Перевірка ефективності розробленої методики.

Дослідження проходило у три етапи:

- констатуючий;
- пошуковий;
- формуючий.

На першому етапі здійснювався аналіз наявної літератури з проблеми дослідження, вивчався досвід роботи викладачів, вчителів розроблялося тематичне та календарно – тематичне планування основи програмування мовою С++. Були проведені опитування, за результатами яких були визначені інтерес учнів до програмування, найбільш пріоритетні для вивчення мови програмування, рівень їх знань і вмінь з програмування. В результаті опитування було встановлено, що рівень пізнавальної активності учнів при вивченні курсу основи програмування є недостатнім. Проте мотивація до його вивчення є досить високою.

На другому етапі були виділені основні аспекти проблеми дослідження, сформована концепція, гіпотеза і завдання дослідження, проаналізовано еволюцію та сучасний стан вітчизняних і зарубіжних курсів програмування,

розкрито можливості об’єктної методології до розв’язування задач мовою програмування С++. Виділення вихідних теоретичних положень дослідження, наявність необхідних експериментальних даних дали змогу організувати і провести пошуковий етап. У ході дослідження було розроблено та прочитано факультативний курс “Основи програмування мовою С++” для учнів 10-х класів. Також на цьому етапі дослідження уточнювалися шляхи індивідуалізації та інтенсифікації процесу навчання, проводився цілеспрямований пошук та відбір змісту навчального матеріалу, його модульний і різнорівневий розподіл. На цьому етапі розроблялася методика реалізації диференційованого підходу у вивченні основ програмування, здійснювалась перевірка ефективності окремих компонентів методичної системи: елементів модульної системи організації навчання, рейтингового контролю знань та вмінь учнів; при цьому досліджувався вплив компонентів на регулярність навчальної діяльності та підвищення рівня самостійної роботи учнів.

На третьому (формуючому) етапі проводилася перевірка ефективності запропонованої методики в процесі навчання основам програмування мовою С++. В процесі дослідження були виділені особливості викладання програмування і відповідні методичні прийоми.

На кожному етапі дослідження аналізувалися одержані результати, вносилися відповідні корективи, уточнювалися вихідні теоретичні положення дослідження і методика. Щоб виключити випадковість в оцінюванні рівня розвитку теоретичних та практичних завдань, у спірних випадках проводилась бесіда, в ході якої учням пропонувалося відповісти на додаткові запитання і виконати завдання, які дають змогу з’ясувати рівень знань.

Програма перевірки ефективності запропонованої нами методики включала:

- облік результатів індивідуалізації та інтенсифікації процесу навчання;

- облік сформованості рівня знань з основ програмування мовою С++ та вмінь розв’язування задач з використанням мови С++.

Провідним методом під час обліку й оцінювання результатів експерименту став метод спостережень за навчальною діяльністю учнів. Одержані дані зіставлялись із результатами бесід, поточного, підсумкового контролю, анкетування.

Опрацювання результатів дослідження та оцінка ефективності запропонованої методики здійснювалася також репродуктивним методом. Не було виявлено статистично значущих відмінностей в експериментальному класі на етапі констатуючого етапу. Враховуючи, що в експериментальному класі був введений змінний фактор – методичні прийоми і засоби навчання основам програмування, спрямовані на розвиток знань та вмінь, можна припустити, що саме це і дало можливість досягти кращих результатів.

Експериментальне навчання засвідчило особливості, характерні для індивідуалізованого процесу навчання, його інтенсифікацію. На перший план у стимулюючо-мотиваційному компоненті навчання вийшли внутрішні та особистісні групи мотивів: ми помітили підвищення пізнавальних потреб учнів, інтересу до навчання, прагнення до самореалізації та прояву потенційних можливостей.

Учні, враховуючи думку вчителя, змогли адекватно проводити самооцінку своїх індивідуальних особливостей, що дозволяло їм самостійно і достатньо обґрунтовано обирати рівень виконання завдань; здійснювати аналіз та прогноз впливу поточного оцінювання на результати підсумкового контролю. Зросла регулярність навчальної діяльності, ініційована не стільки адміністративними вимогами вчителя, скільки суб’єктивними прагненнями учнів.

Результати підсумкового контролю, що включали поряд із анкетування та проведення тестів, дозволили встановити показники рівня знань і вмінь учнів з основ програмування мовою С++ наведено в таблиці 2.5.

Таблиця 2.5

Результати підсумкового контролю

Результати підсумкового контролю				
Рівні досягнень	Початковий рівень (%)	Середній рівень (%)	Достатній рівень (%)	Високий рівень (%)
контрольна підгрупа				
Знання	10	20	40	30
Вміння	0	30	40	30
експериментальна підгрупа				
Знання	0	10	40	50
Вміння	0	0	40	60

Для визначення цих показників нами було проведено контрольні роботи і тести, результати яких подані нижче.

Таблиця 2.6

Результати виконання контрольних робіт учнями експериментальної підгрупи за темами

Результати виконання контрольних робіт учнями експериментальної підгрупи за темами				
Рівні досягнень	Початковий рівень (%)	Середній рівень (%)	Достатній рівень (%)	Високий рівень (%)
Змінні і типи даних в С ++	0	10	30	60
Конструкція розгалуження в С ++	0	20	40	40
Цикли С++	0	10	30	60
Масив в С++	0	20	30	50
Функції в С ++	0	20	40	40
Покажчики в С ++	0	10	50	40
Динамічні масиви в С++	0	20	30	50
Класи в С ++	10	20	40	30
Вектори в С ++	10	10	50	30
Спадкування класів в С ++	0	20	40	40

Рис. 2.8 Результати виконання контрольних робіт учнями експериментальної групи за темами

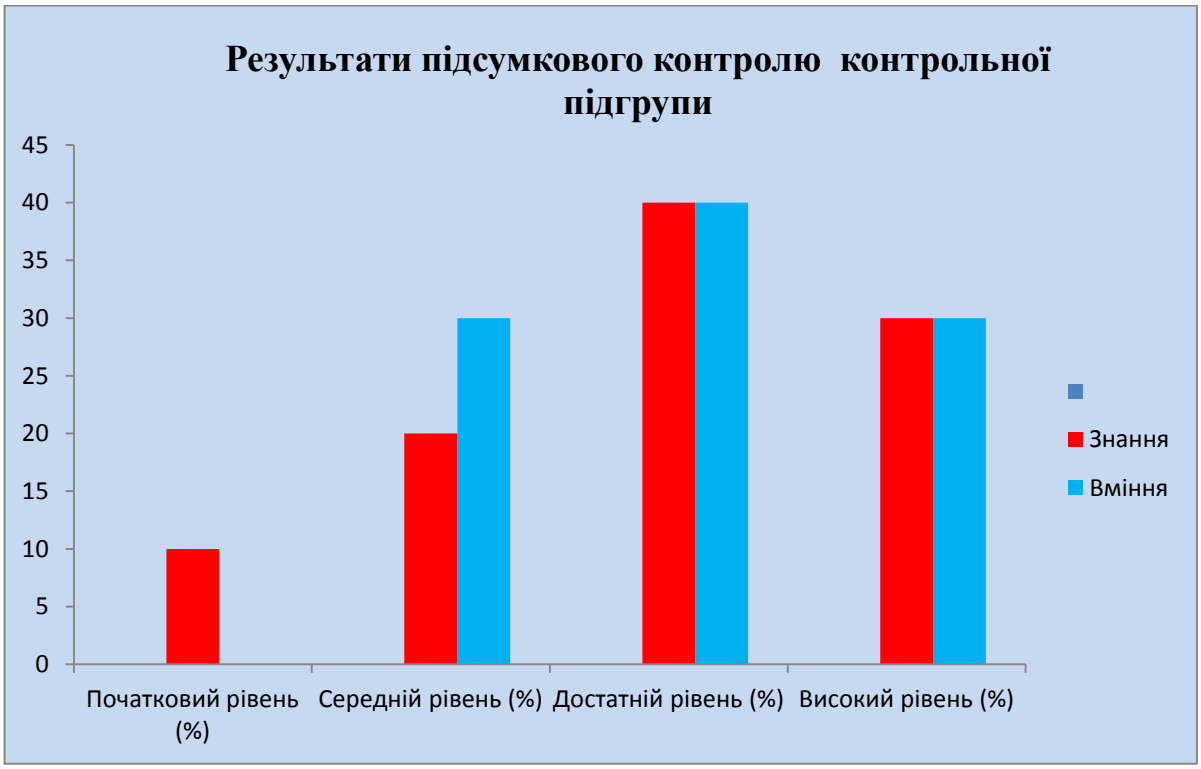

Рис. 2.9 Результати підсумкового контролю контрольної підгрупи

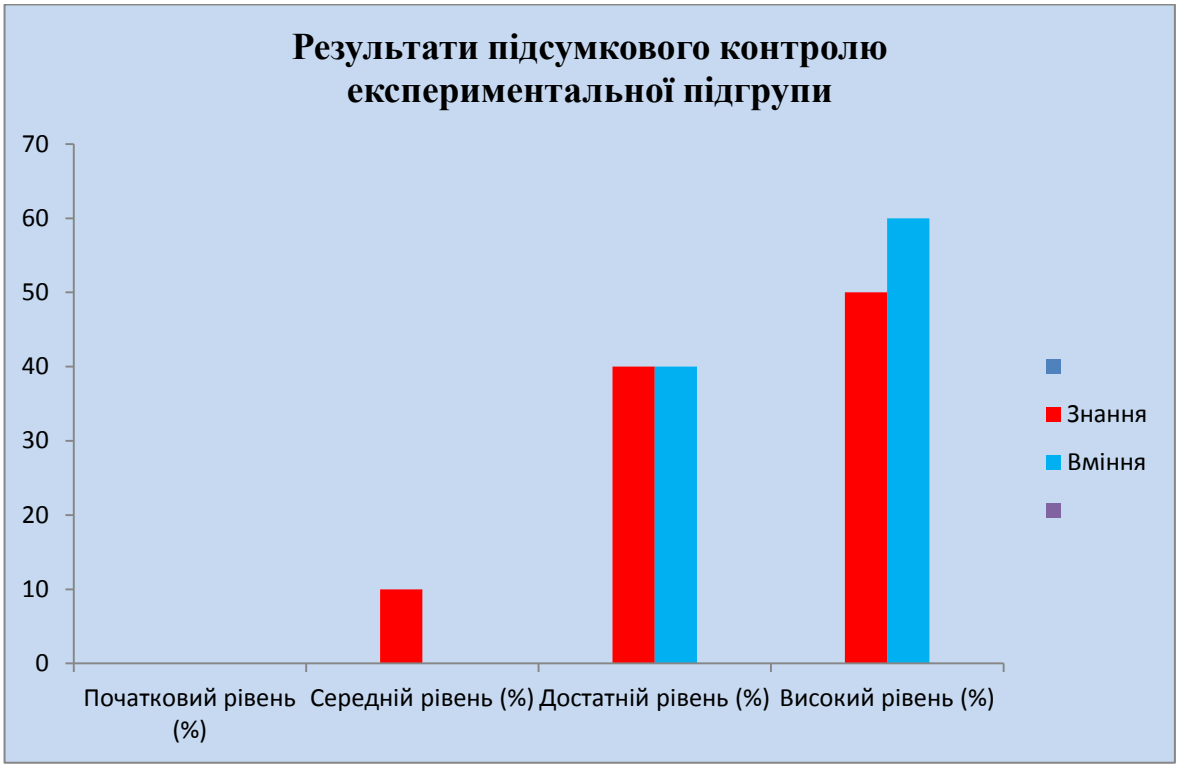

Рис. 2.10 Результати підсумкового контролю експериментальної підгрупи

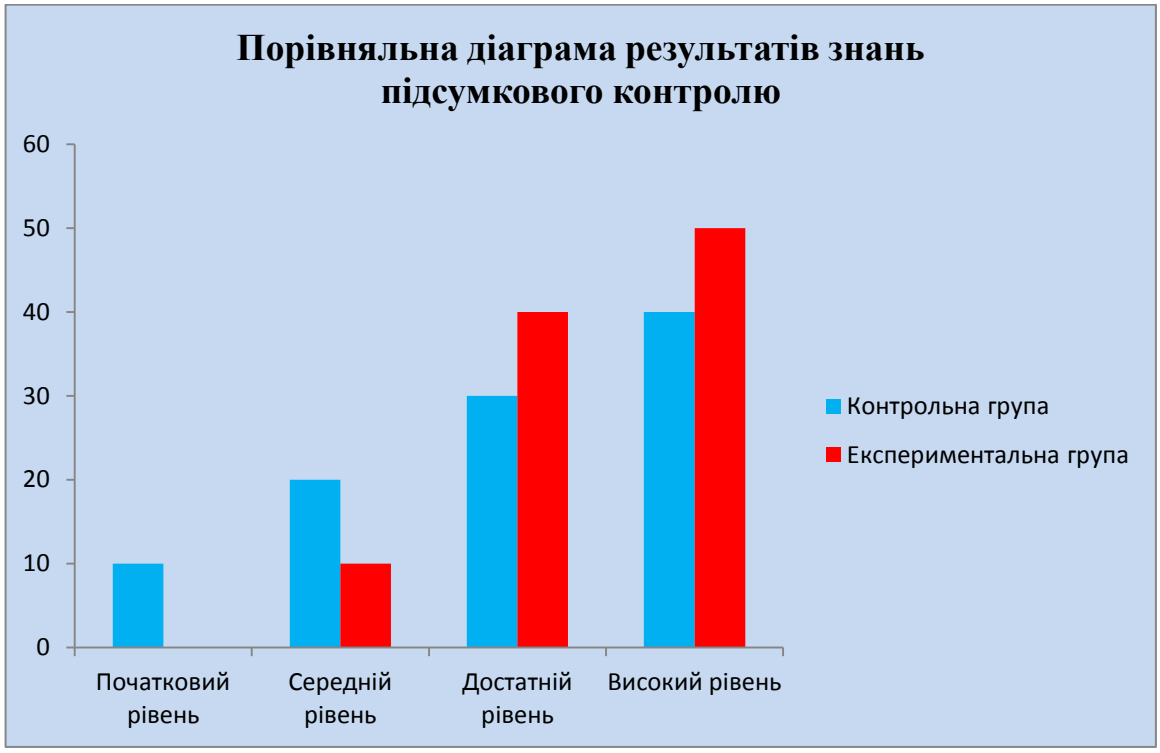

Рис. 2.11 Порівняльна діаграма результатів знань підсумкового контролю

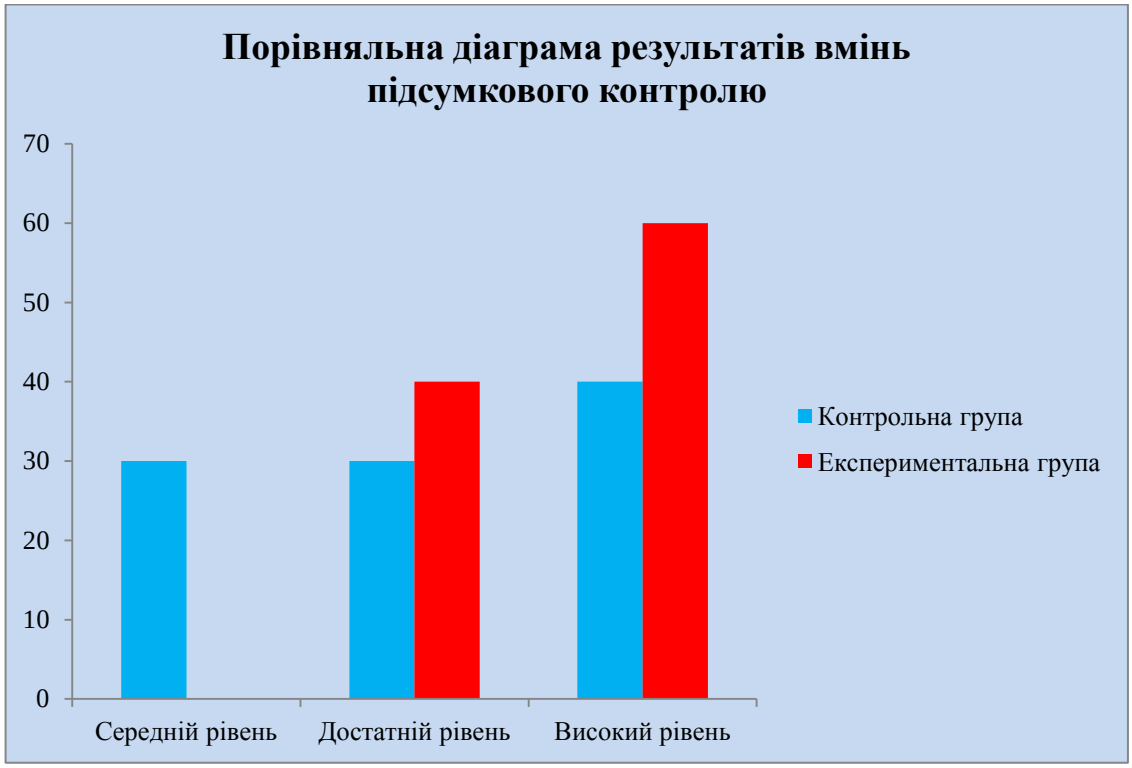

Рис. 2.12 Порівняльна діаграма результатів вмінь підсумкового контролю

У експериментальній групі відмічається незначний розрив між теоретичними знаннями і практичними вміннями учнів контрольної групи. Наведені дані вказують на те, що рівень знань та вмінь, які сформовані в учнів в експериментальній підгрупі (на основі впровадження запропонованої методики), є вищими, ніж в контрольній підгрупі.

Отже, можна говорити, що експеримент виявився вдалим. Аналіз його результатів свідчить про підвищення рівня знань та вмінь у процесі навчання основам програмування мовою С++ , а, отже, і про ефективність розробленої методики.

ВИСНОВКИ ДО РОЗДІЛУ 2

У даному розділі розглянуто методику навчання програмуванню мовою С++ в класах поглибленого вивчення інформатики. Зокрема, розроблено тематичне та календарно-тематичне планування, плани конспекти уроків та практичні роботи. Також розроблено серію контрольних робіт і тестів для перевірки знань учнів. Зроблено аналіз та порівняння результатів контрольного оцінювання. Обрано методи та засоби, які, на нашу думку, найбільш доцільно використовувати під час вивчення основ програмування. Підібрано та розроблено власний сайт для допомоги учням при вивчені основ програмування мовою С++.

В ході експериментального дослідження проведено підсумковий контроль знань і вмінь учнів у контрольній та експериментальній підгрупах. Проведені педагогічні дослідження й аналіз їх результатів дають нам змогу зробити висновок про ефективність розробленої методики і рекомендувати мову С++ для використання у навчальному процесі під час вивчення розділу «Алгоритмізація та програмування» в курсі інформатики 10-х класах, у курсах за вибором, а також у курсі інформатики в класах поглибленого вивчення інформатики.

ВИСНОВКИ

Дипломна робота присвячена питанню методичних особливостей вивчення основ програмування мовою С++ в класах поглибленого вивчення інформатики.

Відповідно до поставлених завдань дослідження у першому розділі дипломної роботи з'ясовано науково-теоретичний і практичний стан проблеми дослідження. Зокрема,тут розглянуто та проаналізовано навчальні програми і підручники з інформатики для закладів загальної середньої освіти. Вибір мови програмування для вивчення основ програмування залишається за вчителем інформатики, тому все більше вчителів обирають мову С++, як одну із найпопулярніших серед розробників, для вивчення основ програмування в класах поглибленого вивчення інформатики або в класах інформаційно-технологічного профілю. Однак сьогодні бракує методичних матеріалів, а тим більше шкільних підручників, для вивчення цієї мови. Недостатньо вивченим є питання методичних особливостей навчання програмуванню мовою С++ в класах поглибленого вивчення інформатики.

У другому розділі дипломної роботи запропоновано методику навчання програмуванню мовою С++ в класах поглибленого вивчення інформатики. Зокрема, розроблено тематичне та календарно-тематичне планування, плани конспекти уроків та практичні роботи. Проаналізовано та обрано методи та засоби найбільш доцільні при вивчені основ програмування. Розроблено власний сайт для організації самостійної навчально-пізнавальної діяльності учнів при вивчені основ програмування мовою С++. Експериментально перевірено ефективність розробленої методики.

Отже, можна констатувати, що на сьогоднішній день існує велика кількість навчальних мов програмування, кожна з яких володіє своїми перевагами. Однак, яку б мову програмування не обрав учитель інформатики для вивчення у середній школі, він має розуміти, що вибір має ґрунтуватися, перш за все, на перспективних, динамічно розвиваючих напрямах науки і

техніки, на розвитку в учнів інженерно-технічного, креативного та логічного мислення.

Результати дипломної роботи можна використати під час вивчення основ програмування в класах поглибленого вивчення інформатики, в класах інформаційно-технологічного профілю, або ж в межах спецкурсу, гуртка чи факультативу з програмування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бак С.М. Застосування методу проектів при вивченні інформатики у класі філологічного профілю / С.М. Бак, Г.М. Ковтонюк // Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми. Збірник наукових праць. – Випуск 17. – Київ-Вінниця: ДОВ „Вінниця”, 2008. – 77-83 с.
2. Бак С.М. Інформатика та обчислювальна техніка. Посібник для студентів 1-х курсів математичних спеціальностей педагогічних ВНЗ. 2-ге вид. / С.М. Бак, Г.М. Ковтонюк. – Вінниця: ПП «Едельвейс і К», 2011. – 448 с.
3. Бак С.М. Інформатика. Частина 1. Посібник для студентів фізико-математичних спеціальностей педагогічних ВНЗ. / С.М. Бак, Г.М. Ковтонюк. – Вінниця: ТОВ «фірма «Планер»», 2012. – 584 с.
4. Барболіна Т.М. Шкільний курс інформатики та методика його викладання: навчальний посіб. / Барболіна Т.М. – Полтава, 2008. – Ч.2. Часткова методика. – 59-66 с.
5. Біляковська О. О. Формування громадянської позиції старшо-класників засобами оцінювання // Вісник Львівського університету. Серія педагогічна / О. О. Біляковська. – Львів : Видавничий центр ЛНУ ім. І.Франка, 2006. – Вип. 21. – Ч. 2. – 141-146 с
6. Бурдун О. В. Інформатика як відображення тенденцій інформатизації освіти І О. В. Бурдун ІІ Науковий часопис НПУ імені М. П. Драгоманова. Серія 2 : Комп'ютерно-оріентовані системи навчання. - 2010. - № 10. - 58-65 с.
7. Галенко СП. Філософські підстави нової парадигми освіти. //Вища освіта; проблеми та перспективи розвитку. - К.; 1995. - 21 - 24 с.
8. Грегори К., Использование Visual C++ 6. Специальное издание. -М.: Вильямс, 2000. -864 с.

9. Ефремова Н. Ф. Тестирование и мониторинг : рекомендации учителю // Стандарты и мониторинг в образовании / Н. Ф. Ефремова. – 2001. – № 3. – 73-75 с.
10. Жалдак М. І. Профільне навчання інформатики І М. І. Жалдак, Н. В. Морзе, О. Г. Кузьмінська II Комп'ютерно-оріентовані системи навчання: збірник наукових праць. - [Вщп. ред. М. І. Жалдак]. - 2004. - Вип. 8. - 13-18 с.
11. Ковтонюк Г. М. Деякі аспекти використання хмарних сервісів у підготовці майбутніх учителів / Г. М. Ковтонюк // Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми. – Випуск 38. – Київ-Вінниця: ТОВ „Планер”, 2014. – 315-319 с.
12. Ковтонюк Г. М. До питання використання комп’ютерного тестування у системі контролю якості знань майбутніх учителів / Г. М. Ковтонюк // Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми. – Випуск 42. – Київ-Вінниця: ТОВ „Планер”, 2015. – 270-273 с.
13. Ковтонюк Г. М. До питання формування інформатичної компетентності майбутніх учителів фізико-математичних дисциплін / Г. М. Ковтонюк // Нова педагогічна думка. – 2017, № 3(91). – 49-51 с.
14. Ковтонюк Г. М. Роль освітніх сайтів у самостійній пізнавальній діяльності школярів та майбутніх учителів / Г. М. Ковтонюк // Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми. – Випуск 24. – Київ-Вінниця: ТОВ „Планер”, 2010. – 44-50 с.
15. Ковтонюк Г. М. Самостійна пізнавальна діяльність школярів як ефективна форма навчальної діяльності / Г. М. Ковтонюк // Наукові записки Вінницького державного педагогічного університету. Серія: Педагогіка і психологія. – Випуск 31. – Вінниця: ТОВ «Планер», 2010. – 71-75 с.

16. Львов М.С., Співаковський О.В. Концепція викладання дисциплін інформатики в школі й педагогічному вузі// Комп'ютер в школі та сім'ї. - 2003. - №3. - 21-25 с.
17. Методичні рекомендації щодо викладання інформатики у 2019/2020 навчальному році [Електронний ресурс]. – Режим доступу до проекту: https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/onovlennya-12-2017/8-informatika.docx.
18. Методичні рекомендації щодо викладання інформатики у 2019/2020 навчальному році [Електронний ресурс]. – Режим доступу до проекту: https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/onovlennya-12-2017/programa-informatika-5-9-traven-2015.pdf.
19. Методичні рекомендації щодо викладання інформатики у 2019/2020 навчальному році [Електронний ресурс]. – Режим доступу до проекту: https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/informatika.pdf.
20. Методичні рекомендації щодо викладання інформатики у 2019/2020 навчальному році. Старша школа (10-11 клас). Рівень стандарту [Електронний ресурс]. – Режим доступу до проекту: https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/informatika-standart-10-11.docx.
21. Методичні рекомендації щодо викладання інформатики у 2019/2020 навчальному році. Старша школа (10-11 клас). Профільний рівень [Електронний ресурс]. – Режим доступу до проекту: https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/01/10-11-profilniy-riven.docx.
22. Морзе Н.В. Методика навчання інформатики. Ч. II Методика навчання інформаційних технологій. – К.: Навчальна книга, 2003. – 288 с.
23. Морзе Н.В. Методика навчання інформатики. Ч. III Основи алгоритмізації та програмування. – К.: Навчальна книга, 2003. – 289 с.

24. Морзе, Н. В.; Барна, О. В. «Інформатика (рівень стандарту)» підручник для 10 (11) класу закладів загальної середньої освіти. Вид.: "Оріон" - 2018. - 240 с.
25. Навчальні програми з інформатики для учнів 10-11 класів загальноосвітніх навчальних закладів (рівень стандарту, академічний рівень, профтьний рівень, поглиблене вивчення) [Електронний ресурс] - Режим доступу до ресурсу: http://mon.gov.ua/activity/education/zagalna-serednya/navchalni-programy.html
26. Основи інформатики та обчислювальної техніки : Програма для середніх закладів освіти І М. І. Жалдак, Н. В. Морзе, Г. Г. Науменко. - К. : 1996. - 12 с.
27. Програма курсу інформатика [Електронний ресурс]. – Режим доступу до проекту: https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/onovlennya-12-2017/programa-informatika-5-9-traven-2015.pdf.
28. Проект концепції розвитку освіти України на період 2015 – 2025 років [Електронний ресурс]. – Режим доступу до проекту: http://www.naiau.kiev.ua/ files/zakon_ukr/proek-rozv-osvitu.pdf.
29. Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А., Шакотько В. В. «Інформатика (рівень стандарту)» підручник для 10 (11) класу закладів загальної середньої освіти. Вид.: "Генеза" - 2018. - 144 с.
30. Саттер, Г. Стандарты программирования на C++: серия "C++ In-Depth" / Г. Саттер, А. Александреску. – М.: Изд. Дом "Вильямс", 2008. - 159 с.
31. Седжвик, Р. Алгоритмы на C++ / Р. Седжвик. – М.: Изд. Дом "Вильямс", 2010. - 96 с.
32. Сергеев А.П., Терен А.Н., Программирование в Microsoft Visual C++ 2005. Самоучитель. -С.Пб.: Диалектика, 2006. -352 с.
33. Солтер Н., Клепер Дж., C++ для профессионалов . -С.Пб.: Диалектика, 2006. -912 с.

34. Співаковський О.В., Львов М.С. Шляхи удосконалення курсу “Основи алгоритмізації та програмування” у педагогічному вузі //Комп’ютер у школі та сім’ї. – 2001. – №4. –22-24 с.
35. Співаковський О.В., Львов М.С. Шляхи удосконалення курсу “Основи алгоритмізації та програмування” у педагогічному вузі //Комп’ютер у школі та сім’ї. – 2001. – №4. – 22-24 с.
36. Хортон А., Visual C++ 2005: базовый курс. -С.Пб.: Диалектика, 2007. - 1152 с.
37. Шелест О.// Науково-популярний альманах «Математика та інформатика навколо нас»/ Вінницький державний педагогічний університет імені Михайла Коцюбинського; [редкол.: М.М.Ковтонюк (голова) та ін.]. – Вінниця: ФОП Рогальська І.О. 2018. – Вип.2. – 302-306 с.
38. Шелест О.// Науково-популярний альманах «Математика та інформатика навколо нас»/ Вінницький державний педагогічний університет імені Михайла Коцюбинського; [редкол.: М.М.Ковтонюк (голова) та ін.]. – Вінниця: ФОП Рогальська І.О. 2019. – Вип.3. – 289-292 с.
39. Шилдт Г. , Справочник программиста по C/C++, 3-е издание. -М.: Вильямс, 2003. -432 с.
40. Шилдт, Г. C++: методики программирования Шилдта / Г. Шилдт. – М.: Изд. Дом "Вильямс", 2008. - 137 с.

ДОДАТКИ

ДОДАТОК А

План – конспект уроку в 10 класі

Тема уроку 1. Поняття мови програмування. Складові мови програмування

Мета:

- освітня: ознайомити учнів з поняттям мови програмування, основними складовими мови програмування;
- розвивальна: розвивати логічне та алгоритмічне мислення, вміння діяти за інструкцією, збагачувати мовлення учнів, аналізувати і робити висновки;
- виховна: виховувати інформаційну культуру учнів, дбайливе ставлення до шкільної комп'ютерної техніки.

Тип уроку: Урок засвоєння нових знань

Обладнання: плакат "Інформатика", стенди "Вивчаємо тему", "Критерії оцінювання", "Мова програмування"

Хід уроку

I. Організаційний етап

Вітання з класом. Перевірка присутності і готовності учнів до уроку.

II. Мотивація навчальної діяльності.

III. Актуалізація опорних знань

Фронтальне опитування:

1. Де у сьогоднішній час використовують комп`ютери і які завдання вони виконують?
2. Які програмні засоби використовуються для виконання цих завдань?
3. Чи хотіли б ви створювати програми для вирішення своїх завдань?

IV. Повідомлення теми, цілей, завдань уроку.

Мета сьогоднішнього уроку ознайомитись з мовами за допомогою яких створюються програми, їхньою класифікацією та особливостями, вивчити призначення мов програмування та основні складові мов програмування.

V. Сприймання і усвідомлення учнями нового матеріалу.

Пояснення вчителя з елементами демонстрування (використовуються можливості локальної мережі класу або проектор).

Комп`ютер не може виконувати команди, які зрозумілі для людей, тому потрібно його навчити це робити або використати перекладач. При створені програм програмісти виконують роль перекладачів із звичної нам мови на мову комп`ютера. Отже, мова програмування це формальна мова, що забезпечує зручний опис конкретних проблем, які формулюються людиною й розв'язуються за допомогою комп'ютера.

Мови класифікують за такими критеріями:

- Рівень абстракції . Мови програмування високого рівня оперують сутностями ближчими людині, такими як об'єкти, змінні, функції. Мови програмування нижчого рівня оперують сутностями ближчими машині: байти, адреси, інструкції. Текст програми на мові високого рівня зазвичай набагато коротший ніж текст такої самої програми на мові низького рівня, проте програма має більший розмір.
- Область застосування. Універсальні та спеціалізовані. Універсальні мови використовуються для вирішення різних завдань. Спеціалізовані мови призначені для вирішення завдань одного, максимум кількох, видів завдань. Наприклад, роботи з базами даних, web-програмування або написання скриптів для адміністрування операційних систем.

Підтримувані парадигми програмування:

- Об'єктно-орієнтовані, логічні, функційні, структурні
- Об'єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування (ООП) — це технологія створення складного програмного забезпечення, яке засноване на представленні програми у вигляді сукупності об'єктів, кожен з яких є екземпляром певного класу, а класи утворюють ієрархію із спадкоємством властивостей.

Мови програмування створюються, як система позначень для створення алгоритмів та типів даних. Кожна мова програмування має свої правила написання та управління.

Основними елементами мови програмування є:

- символи;
- слова;
- вирази;
- команди (оператори).

Символи мови - це основні нероздільні знаки, за допомогою яких описуються програми і дані.

Слова мови - структури, що утворені із символів алфавіту мови програмування і мають певний зміст. Слова - це імена змінних та констант, числа, службові слова та ін.

Вираз - це текст, що задає правило обчислення одного значення величини. Якщо одержуване значення числове, то вираз називають арифметичним, якщо значення логічне, то вираз називають логічним або бульовим, якщо одержуване значення - текст, то вираз називають літерним.

Команда - це вказівка про виконання деякої дії. При написанні програм команди називають операторами, а величини, що використані в команді - операндами.

Синтаксис мови програмування визначає те, як буде виглядати програма на цій мові, зокрема, як пишуться оператори, оголошення і інші мовні конструкції.

Тип даних — це деякий клас об'єктів даних разом з набором операцій для створення і роботи з ним.

Основні мови програмування:

- C - стандартизована процедурна мова програмування. Створена для використання в операційній системі UNIX..
- C++ - компільована статично типізована мова програмування загального призначення.
- Free Pascal - вільно розповсюджуваний компілятор мови програмування Pascal.
- Java - об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems.
- C# відноситься до сім'ї мов із C-подібним синтаксисом, з них її синтаксис найбільш близький до C++ і Java.
- PHP - скриптова мова програмування загального призначення, інтенсивно застосовується для розробки веб-додатків.
- ActionScript - об'єктно-орієнтована мова програмування, яка додає інтерактивність, обробку даних і багато чого іншого до вмісту Flash-додатків.
- Perl - високорівнева динамічна мова програмування загального призначення, що інтерпретується.
- Python - високорівнева мова програмування загального призначення з акцентом на продуктивність розробника і читаність коду.
- Ruby - динамічна, рефлективна, інтерпретована високорівнева мова програмування для швидкого і зручного об'єктно-орієнтованого програмування.

Осмислення, узагальнення і систематизація набутих знань.

Робота в зошитах.

Опорний конспект

<table>
<tr><td colspan="10">Мова програмування</td></tr>
<tr><td colspan="10">формальна мова, що забезпечує зручний опис конкретних проблем,
які формулюються людиною й розв'язуються за допомогою комп'ютера.</td></tr>
<tr><td colspan="10">Мови програмування</td></tr>
<tr><td>C</td><td>Ruby</td><td>C++</td><td>Perl</td><td>Python</td><td>Free Pascal</td><td>Java</td><td>C#</td><td>PHP</td><td>ActionScript</td></tr>
<tr><td colspan="10">Елементи мови програмування</td></tr>
<tr><td colspan="2">символи</td><td colspan="2">слова</td><td colspan="2">вирази</td><td colspan="2">команди</td><td colspan="2">синтаксис</td><td>типи даних</td></tr>
</table>

Виконання комплексу вправ для зняття зорової втоми.

Учитель, враховуючи індивідуальні особливості учнів класу, самостій но визначає час і термін проведення комплексу вправ під час роботи (як правило, через 8-10 хвилин після початку роботи).

VI. Домашнє завдання:

1. Опрацювати конспект уроку.

2. Додаткове завдання.

Підготувати 1-2 запитання з теми, що вивчається.

VII. Підсумок уроку.

Рефлексія.

Учням пропонується закінчити речення:

- "Для мене сьогодні важливим було...",
- "Сьогодні я дізнався про...",
- "Мені хотілося в майбутньому дізнатись про..., навчитись …".

IX. Оцінювання роботи учнів на уроці.

ДОДАТОК Б

Практична робота № 2

Тема: Базові навички роботи з середовищем С++

Мета: ознайомитися з середовищем мови програмування Borland C++ Builder 6, реалізувати програму для найпростіших дій введення та виведення інформації.

Завдання:

1. Запустити середовище програмування *C++ Builder 6 (Пуск->Програми->Borland C++ Builder 6->C++ Builder 6)*

2. Перегляньте всі пункти меню: File, Edit, Search, View, Project, Run, Component, Database, Tools, Window, Help. В звіті вкажіть їх призначення.

3. Для роботи в консольному режимі C++ Builder необхідно виконати наступні кроки

4. Збережіть файл натиснувши File -> Save as…, під ім’ям lr1.cpp.

5. Наберіть в робочій частині вікна наступні команди:

```
#include <vcl.h>
#include <iostream.h>
int main() {
cout<< "Hello C++\n";
}
```

6. Запустить програму на виконання, обравши команди Run->Run або F9. Програма виконала дію і зник консоль.

7. Для того щоб бачити виконану дію програми підключимо до розділу об’яви бібліотек директиву для роботи з екраном #include <conio.h> та введемо функцію для затримки екрана до натиснення будь-якої клавіши: getch(); тепер програма має вигляд:

```
#include <vcl.h>
#include <iostream.h>
#include <conio.h>
int main() {
```

```
cout<< "Hello C++\n";
getch();
}
```

8. Запустіть програму на виконання, на чорному екрані з’явиться повідомлення. В звіті вкажіть яке? Далі натисніть Enter або Esc для переходу до програми.

9. Модифікуймо нашу програму до вигляду:

```
#include <vcl.h>
#include <iostream.h>
#include <conio.h>
int main()
{
char name[20];
cout<< "What is your name:\n";
cin >> name;
cout<< "Hello: " <<name << endl;
getch();
}
```

Після запуску вона повинна вивести на екрані питання *What is your name:*, ми відповідно повинні ввести ім’я, наприклад *Ivan* та натиснути клавішу Enter, на це програма повинна вивести: *Hello: Ivan.*

10. Запишіть в звіті модифіковану програму з коментарем біля кожного рядка програми.

Наприклад:

```
#include <iostream.h> //підключення бібліотеки для введення-виведення інформації
```

Контрольні питання:

1. З чого складається алфавіт мови С++?
2. Які службові слова використовує мова С++?
3. Що являє собою структура програми?

4. Наведіть основні вимоги, які слід ураховувати при створенні програм мовою C++.

ДОДАТОК В

Для організації навчальної діяльності учнів в експериментальній групі нами було розроблено сайт https://sites.google.com/view/shelestoksana-vdpu. та блог http://oksanashelest.blogspot.com. На цьому сайті розміщено теоретичний матеріал, практичні завдання, тести.

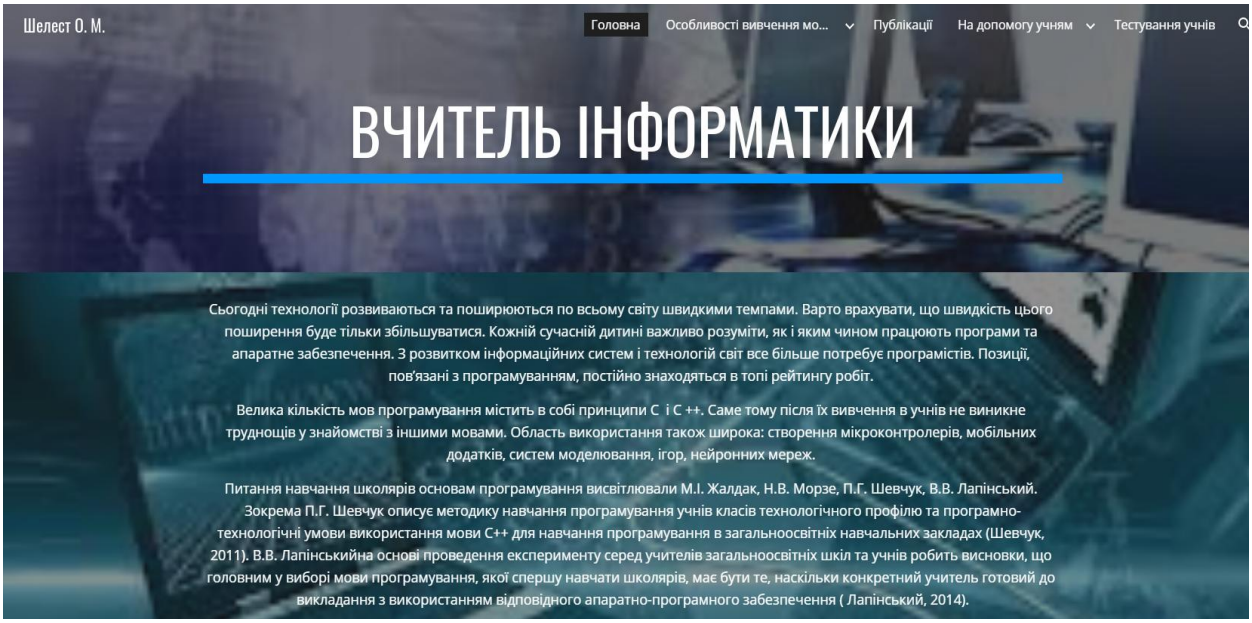

Рис. В.1 Розроблений сайт

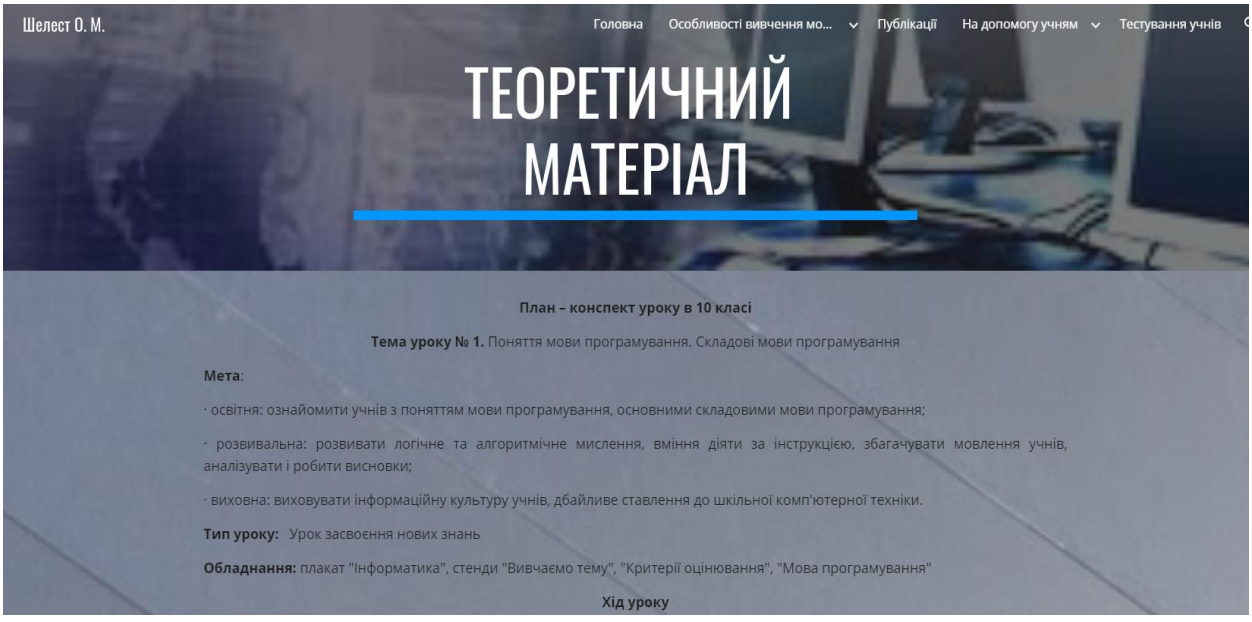

Рис. В.2 Розроблений сайт

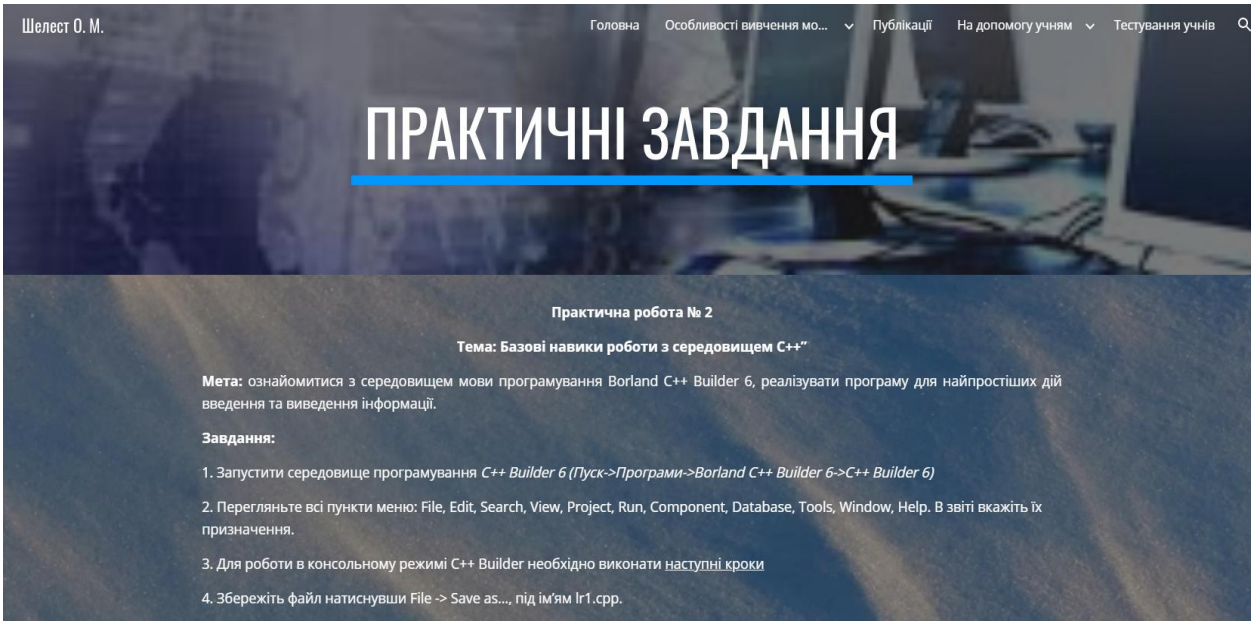

Рис. В.3 Розроблений сайт

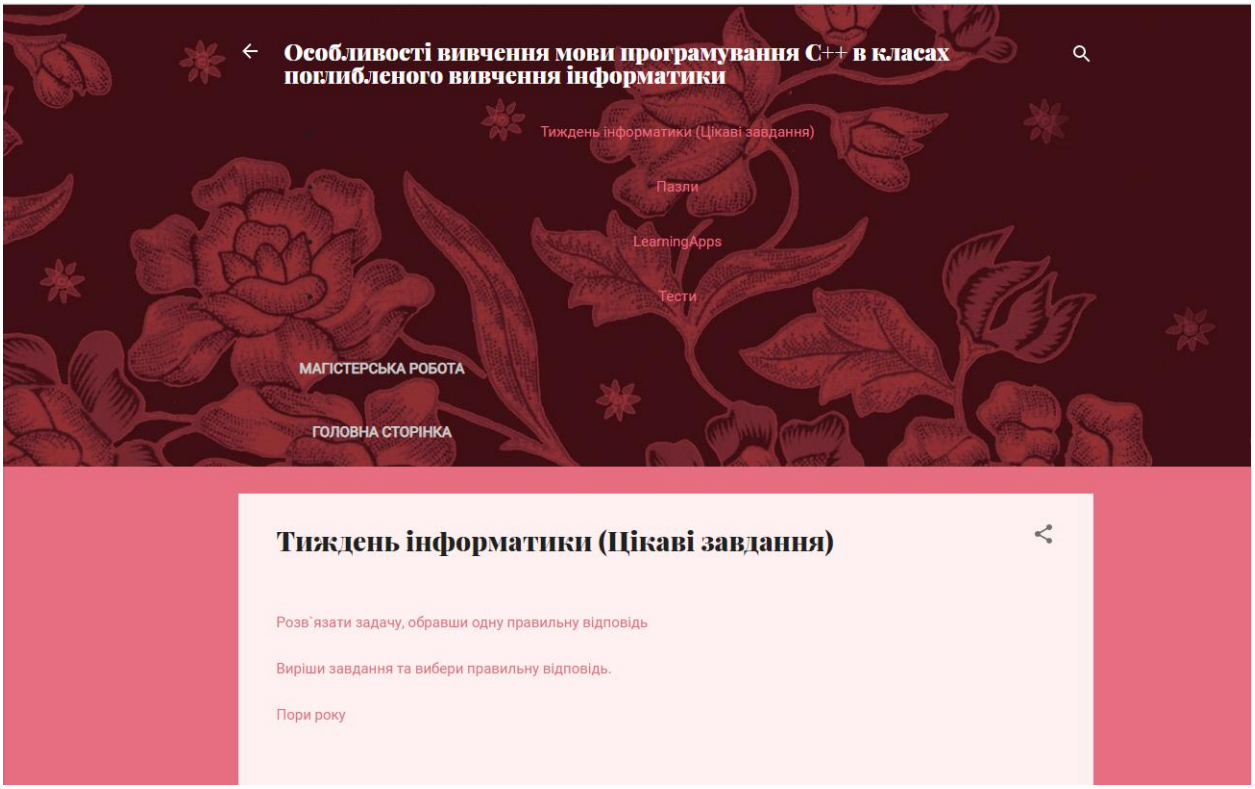

Рис. В.4 Розроблений блог

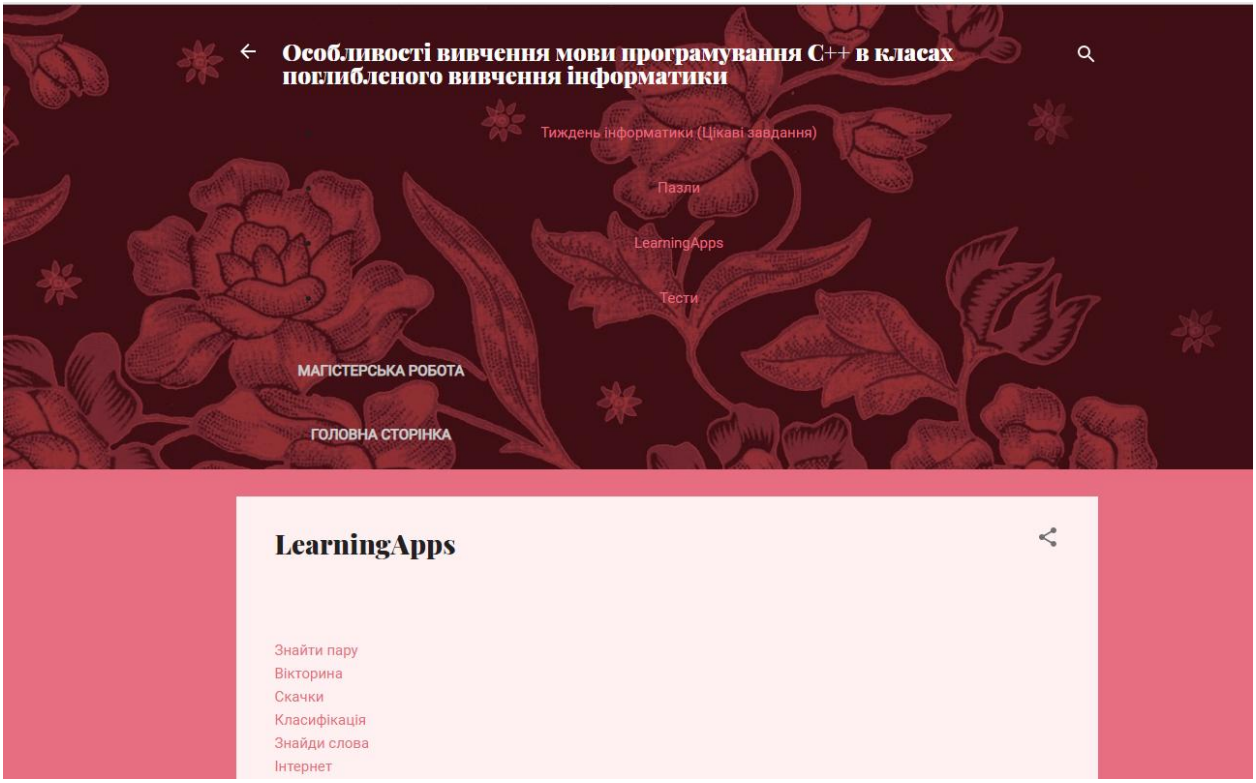

Рис. В.5 Розроблений блог