

Навчальна програма з позашкільної освіти
науково-технічного напрямку
інформаційно-технічного профілю
«СИСТЕМНЕ ПРОГРАМУВАННЯ»
1 рік навчання

Чебанюк О. В., керівник гуртка лабораторії комп'ютерних технологій
відділу науково-технічної творчості
Київського Палацу дітей та юнацтва;

Белоглазова Н. В., начальник відділу змісту позашкільної освіти
відділення змісту позашкільної освіти та виховної роботи
Інституту інноваційних технологій і змісту освіти

ПОЯСНЮВАЛЬНА ЗАПИСКА

Актуальність навчальної програми пояснюється тим, що інформаційні технології (ІТ) зараз необхідні в усіх галузях діяльності людини, і попит на фахівців з ІТ постійно зростає.

Особливість даної навчальної програми полягає в тому, що досить складні для учнів технології розробки системного програмного забезпечення необхідно викладати цікаво — за допомогою творчих завдань з використанням мови C++. Серед сучасних мов програмування найбільшою та такою, що динамічно розвивається, є «родина» C-подібних мов (C++, C# , Java). Популярність мови C# швидко зростає, і, як альтернатива мові Java, вона вміщує в собі найкращі досягнення Java, C, C++ і Delphi.

Організація навчального матеріалу здійснюється за лінійним принципом: від простого до складного. Увага акцентується на розвитку дослідницьких умінь, пошуку власних способів виконання робіт.

Під час підготовки програми були враховані Закон України «Про основні засади розвитку інформаційного суспільства в Україні на 2007–2015 роки», Закон України «Про національну програму інформатизації», Положення про порядок організації індивідуальної та групової роботи в позашкільних навчальних закладах (у редакції наказу Міністерства освіти і науки України від 10.12.2008 р. № 1123) та інші нормативні документи.

Головна мета програми «Системне програмування» — засвоєння вмінь і навичок, необхідних для розробки програм мовою C++ для використання функцій системного програмного забезпечення.

Основні завдання програми:

- сформувати практичні навички у використанні мови програмування C++ для розв'язання прикладних задач;

- навчити самостійно розробляти програми (ставити задачі, створювати математичну та логічну моделі, проектувати алгоритм методом покрокової деталізації, створювати та налагоджувати програми).

За програмою передбачено навчання протягом року групи з 12–15 вихованців віком 15–17 років. Кількість годин — 144 на рік (4 години щотижня).

Програма орієнтована переважно на практичну творчу роботу з використанням комп'ютера.

Основною формою проведення занять є групова. Практичну роботу виконують вихованці як індивідуально, так і в групах, відповідно до методики, яку обрав керівник гуртка і яка передбачає виконання конкретного однакового для всіх завдання.

Навчально-тематичний план програми є орієнтовним. Керівник гуртка може змінювати кількість теоретичних і практичних занять (залежно від того, як швидко та якісно вихованці засвоюють теоретичні знання та набувають практичних навичок), враховуючи рівень інтелектуального розвитку гуртківців, математичної підготовки тощо. Різницю розподілу навчальних годин вказують у календарно-тематичному плані навчання.

Знання та вміння контролюють протягом усього періоду навчання, оцінюючи виконання практичних і самостійних завдань, запланованих відповідно до програми, а також під час участі гуртківців у конкурсах, змаганнях та олімпіадах. Основними способами контролю ефективності навчання можуть бути: тестування, співбесіда, система заліків, контрольні завдання, захист власних проєктів, а також іспит. Формуючи групу, слід враховувати вікові особливості дітей і зміст програми. Учні зараховують за результатами тестування, оскільки для опанування програми «Системне програмування» потрібно мати певний рівень знань і вмінь користувача-початківця.

Вихованці розробляють програми з блоками перевірки того, що значення параметрів належать до області допустимих значень, і що при виконанні програми не буде аварійних зупинок. Вони вивчають способи введення та зміни даних таким чином, щоб вони були зручними під час користування програмним продуктом, а форма подання результатів такою, що дозволяє швидко пересвідчитися в правильності розрахунків. Разом із умінням обирати найраціональніший алгоритм та обґрунтовувати свій вибір учні вчаться оформлювати блочні структури, бути економними в обчисленнях і використанні пам'яті.

Програмне забезпечення, необхідне для вивчення запланованого навчального матеріалу:

- операційна система Windows XP, Windows 7 або Linux;
- інтегроване середовище розробки (IDE) програм мовою C++ (Microsoft Visual Studio 10, Microsoft Visual Studio Express, або Borland C++ Builder);
- програми архівування даних для роботи з архівними файлами;
- набір програм для роботи в інтернеті для перегляду web-сторінок, роботи з електронною поштою та телеконференціями, файловими архівами, пошуковими системами тощо.

Для якісного навчання 12–15 вихованців слід забезпечити кожного індивідуальним робочим місцем. Обов'язковою умовою є належний технічний стан комп'ютерів (достатній обсяг оперативної пам'яті та жорсткого диску, наявність DVD-ROM, рознімів для USB флеш-пам'яті; доступ до інтернету). Комп'ютери повинні бути об'єднані в локальну мережу і мати необхідне програмне забезпечення, а також периферійні пристрої. Навчальний процес слід супроводжувати наочним і дидактичним матеріалом (спеціальною методичною літературою, інтернет-ресурсами, освітньо-інформаційними ресурсами тощо).

Основний рівень, один рік навчання

НАВЧАЛЬНО-ТЕМАТИЧНИЙ ПЛАН

Розділ, тема	Кількість годин		
	теоретичних	практичних	усього
Вступ	2	–	2
Розділ 1. Основи мови програмування C++	34	44	78
1.1 Основи алгоритмізації	2	4	6
1.2. Базові поняття C++	4	4	8
1.3. Оператори та вирази	4	4	8
1.4. Цикли	4	8	12
1.5. Процедури. Функції користувача	8	8	16
1.6. Масиви, рядки, файли. Управління пам'яттю	12	16	28
Розділ 2. Об'єктно-орієнтоване програмування	24	38	62

2.1. Класи. Властивості та методи класів	8	10	18
2.2. Наслідування, поліморфізм	8	10	18
2.3. Компоненти. Створення компонентів	8	10	18
2.4. Розробка проекту	–	8	8
Підсумок	–	2	2
Разом	60	84	144

ЗМІСТ ПРОГРАМИ

Вступ (2 год.)

Теоретична частина. Сучасні професії в галузі інформаційних комп'ютерних технологій. Вимоги до фахівців ІТ. Вступний інструктаж з безпеки життєдіяльності вихованців.

Розділ 1. Основи мови програмування C++ (78 год)

1.1. Основи алгоритмізації (6 год.)

Теоретична частина. Поняття алгоритму. Властивості алгоритму. Способи запису алгоритмів. Блок-схеми. Поняття програми. Поняття про мови програмування. Класифікація мов програмування. Формальна мова запису алгоритмів. Алгоритми лінійної, розгалуженої та циклічної структури. Внутрішня будова комп'ютера, принцип роботи програм.

Практична частина. Побудова блок-схем алгоритмів лінійної, розгалуженої та циклічної структури.

1.2. Базові поняття C++ (8 год.)

Теоретична частина. Середовище програмування Borland C++ Builder. Запуск і робота. Інтерфейс середовища програмування Borland C++ Builder. Створення нового файлу, відкриття наявного, збереження. Вікно редагування коду. Запуск і налагодження додатку.

Основні елементи мови C++. Алфавіт. П'ять груп символів. Константи, змінні, керівні послідовності, ідентифікатори, ключові слова, коментарі. Типи даних. Структура програми на C++ (функція main, директива #define секції include тощо). Виконання програми. Компіляція.

Практична частина. Створення найпростішого консольного застосування мовою C++.

1.3. Оператори та вирази (8 год.)

Теоретична частина. Оператор присвоювання. Математичні оператори +, –, *, / , %. Математичні функції (бібліотека <math.h>). Особливості

арифметичних операцій. Пріоритети операцій. Інкремент/декремент. Формати виведення даних.

Логічні оператори. АБО (||), І (&&), заперечення (!). Умовний оператор if...else.

Практична частина. Створення та налагодження програми, що виконує математичні операції.

1.4. Цикли (12 год.)

Теоретична частина. Призначення та загальні поняття циклів. Цикл з відомим числом кроків (for). Цикл з передумовою (while). Цикл з постумовою (do — while). Оператор switch ...case.

Перетворення типів (явне та неявне).

Практична частина. Створення та налагодження програми, що реалізує циклічні алгоритмічні структури.

1.5. Процедури. Функції користувача (16 год.)

Теоретична частина. Поняття процедури. Поняття функції. Відмінність процедури від функції. Тип void. Область видимості змінних. Локальні та глобальні змінні.

Передавання параметрів за посиланнями. Випадкові числа. Функції для роботи з випадковими числами rand, random, srand, randomize. Рекурсія. Робота в графічному режимі. Графічні режими екрана. Графічні функції. Обробка подій клавіатури.

Практична частина. Виконання індивідуальних завдань з використанням власних функцій.

1.6. Масиви, рядки, файли. Управління пам'яттю (28 год.)

Теоретична частина. Робота з масивами. Простий пошук у масиві. Формування масиву за заданою умовою. Перестановка елементів у масиві. Сорткування елементів масиву. Масиви в процедурах і функціях. Використання символьних рядків. Основні функції роботи з рядками. Порівняння, копіювання, об'єднання, форматування символьних рядків.

Обробка файлів периферійних пристроїв: обмін з файлами послідовного доступу, обмін даними з бінарними файлами, обмін даними з телекомунікаційними файлами. Функції C++ для роботи з файлами.

Управління пам'яттю. Показчики та посилання. Динамічне виділення пам'яті. Поняття структури. Структури як параметри функцій. Передавання параметрів за посиланням.

Практична частина. Створення програм, що виконують обробку текстових і бінарних файлів.

Розділ 2. Об'єктно-орієнтоване програмування (62 год.)

2.1. Класи. Властивості та методи класів (18 год.)

Теоретична частина. Основні поняття об'єктно-орієнтованого програмування (ООП). Класи й об'єкти. Основні властивості об'єкту. Інкапсуляція (об'єкт у вигляді «чорного ящика»). Модифікатори доступу (private, public, protected). Створення, життя та смерть об'єктів. Конструктори та деструктори.

Практична частина. Створення власного класу. Розробка та налагодження програми, що використовує цей клас.

2.2. Наслідування, поліморфізм (18 год.)

Теоретична частина. Поняття похідного та базового класів. Відношення наслідування між класами. Показники базових класів. Множинне наслідування. Поліморфізм. Віртуальні методи. Раннє та пізнє зв'язування. Абстрактні класи. Поняття компонентів. Створення власних компонентів. Масиви об'єктів. Посилання.

Практична частина. Передача параметрів конструктору базового класу. Перевизначення методів базового класу. Створення діаграми класів.

2.3. Компоненти. Створення компонентів (18 год.)

Теоретична частина. Поняття компонентів. Палітра компонентів Borland C++ Builder. Компоненти для роботи з операційною системою. Створення власних компонентів.

Практична частина. Розробка застосувань з використанням стандартних і власних компонентів.

2.4. Розробка проекту (8 год.)

Практична частина. Розробка власного проекту за темами:

- «Розробка програми складання лабіринтів»;
- «Розробка програми пошуку шляху в лабіринті»;
- «Розробка системи моделювання поведінки абонентів під час обрання засобів зв'язку»;
- «Розробка програмної моделі взаємодії абонентів з бібліотекою»;
- «Моделювання процесу прокладання маршруту дорогами»;
- «Моделювання процесу обрання та купівлі товарів у магазині самообслуговування»;
- «Моделювання процесу керування автомобілем на шосе з кількома смугами»;
- «Моделювання пересування вантажів за допомогою блоків і тросів».

Підсумок (2 год.)

Підбиття підсумків роботи.

ПРОГНОЗОВАНІ РЕЗУЛЬТАТИ

Вихованці мають знати і розуміти:

- призначення й основні функції інтегрованого середовища Borland C++ Builder;
- правила роботи з інтегрованим середовищем Borland C++ Builder;
- операції, які можна виконувати з інтегрованим середовищем Borland C++ Builder;
- правила розробки об'єктів;
- принципи побудови ієрархії об'єктів;
- технологію створення класів;
- правила розробки графічних і мультимедійних додатків;
- тонкощі програмування під Windows і використання API функції;
- основні відомості про реляційні бази даних;
- компоненти для роботи з базами даних;
- принципи технології client/server.

Вихованці мають вміти і застосовувати:

- працювати з object inspector;
- редагувати object event;
- використовувати властивості та події Mplayer;
- розробляти структуру одноярусної бази даних;
- виконувати запити до бази даних;
- змінювати конфігурацію панелі компонентів;
- створювати об'єкт відповідно до правил структуризації програм;
- створювати власні класи компонентів.

Вихованці мають набути досвід:

- складання програм різного рівня складності;
- налагоджувати програми (TD та CPU). Використовуючи відповідний інструментарій;
- розробки одноярусної бази даних;
- використання об'єкта GDI для роботи з графікою.

ЛІТЕРАТУРА

1. Ватсон Б. C# 4.0 на прикладах. — М.: Вільямс, 2010. — 610 с.
2. Герберт Шилдт. C++: базовий курс (C++ from the Ground Up) – 4-е изд. — М.: Вільямс, 2008. — 624 с.

3. Дрейер М. С# для школьников. — М.: БИНОМ, Лаборатория знаний, 2010. — 128 с.
4. Закон України «Про освіту» // Освіта України. — 1996. — № 123.
5. Закон України «Про позашкільну освіту» // Урядовий кур'єр. — 2000. — № 25.
6. Ковалюк Т. В. Основи програмування. — К.: Видавнича група ВНУ, 2005. — 384 с.
7. Культин Н. С/C++ в задачах и примерах. — СПб.: БХВ-Петербург, 2009. — 368 с.
8. Лехан С. А. Інформатика. Мова програмування C++. Спецкурс. — Шепетівка: Аспект, 2007. — 159 с.
9. Мак-Дональд М. Silverlight 3 с примерами на С# для профессионалов. — М.: Вильямс, 2010. — 560 с.
10. Мартынов Н. Н. С# для начинающих. — М.: КУДИЦ-ПРЕСС, 2007. — 272 с.
11. Павловская Т. А. С/C++. Программирование на языке высокого уровня. СПб.: Питер, 2007. — 461 с.
12. Положення про позашкільний навчально-виховний заклад // Інф. зб. Міністерства України. — 1994. — № 9.
13. Програми з позашкільної освіти: науково-технічний напрям / Биковський Т. В., Вихренко Т. О. та ін. — К., 2012.
14. Рейтинг языков программирования - июнь 2011 [Електронний ресурс] сайт: «Studlab.com – студенческая лаборатория». — Режим доступа: http://studlab.com/news/rejting_jazykov_programmirovanija/2011-06-09-125