Документ подписан ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Владелец: Головкина Татьяна Владимировна

Должность: Директор

Дата подписи: 01.09.2021

**МБОУ Гимназия №91 имени М.В.Ломоносова**

|  |  |  |
| --- | --- | --- |
| **Рассмотрено:**  **На заседании кафедры**  **Протокол №\_\_\_\_**  **«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_\_г.** | **Согласовано:**  **На НМС**  **Протокол №\_\_\_\_**  **«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_\_г.** | **Утверждено:**  **Директор МБОУ Гимназия №91 им. М.В.Ломоносова**  **\_\_\_\_\_\_\_\_\_\_\_\_Головкина Т.В.**  **«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_\_г.** |

**ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА**

**дополнительного образования**

**«Программирование»**

для учащихся 5-11-х классов

**Срок реализации**: 1 год

**Направленность**: техническая

**Составитель:**

педагог дополнительного образования  
Шевчугов В.О.

Железногорск – 2021 г.

**Пояснительная записка программе «Программирование»**

**Направленность программы**: техническая.

**Новизна, актуальность, педагогическая целесообразность:** Программирование как тема курса информатики, с одной стороны, и как профессиональная деятельность, с другой стороны, в информационном обществе приобретает все большее значение. Небольшой объем часов в курсе школьной информатики, выделяемый на изучение данного раздела не достаточен. При этом потребности общества, а также проводимые олимпиады всех уровней: от школьного до международного, с узкой направленностью на программирование требуют учащихся все более развитых навыков программирования.

Курс «Программирование» позволит учащимся, познакомиться с основами программирования на языке высокого уровня С++, создать задел для обучения объектно-ориентированному программированию, что должно в дальнейшем облегчить изучение других высокоуровневых языков. Одновременно с этим, есть возможность выявить ребят, которые хорошо логически мыслят и уже готовы к решению простейших задач связанных с разработкой собственных приложений, а это позволит в дальнейшем подготовить их к программированию и, возможно, определит их будущий профиль обучения.

**Цель программы обучения**:

Обучение школьников азам объектно-ориентированного программирования.

**Задачи программы обучения:**

* развитие у учащихся абстрактного, логического и алгоритмического мышления;
* индивидуализация процесса образования посредством дифференцирования заданий по уровню сложности и объему, что призвано обеспечить эффективность самостоятельной работы учащихся;
* обучение основам моделирования и программирования, выявление программистских способностей школьников;
* развитие познавательных интересов, интеллектуальных и творческих способностей учащихся.

**Отличительные особенности дополнительной образовательной программы**: Рабочая программа по курсу «Программирование» для 5-11 класса направленна преимущественно на практическое знакомство учащихся с программированием. Более чем две трети программы отведено самостоятельной работе. В программе сделан уклон на разработку программ под операционную систему Windows, которая наиболее близка учащимся.

**Возраст детей, участвующих в реализации данной программы**: 12-17 лет (5-11 классы)

**Срок реализации программы**: 1 год

**Формы и режим занятий**: Единицей учебного процесса является урок, включающий в себя объяснение нового материала и выполнение практических заданий в тетради или на компьютере, направленных на закрепление изученного материала. Занятия проходят 1 раз в неделю, по 4 часа.

**Ожидаемые результаты и способы определения их результативности**:

*Учащиеся должны знать/понимать:*

- синтаксис основных операторов языка;

- типы данных;

- понимать принципы преобразования типов данных.

- понимать принципы выделения памяти из «кучи»;

- понимать способы хранения и адресации данных в оперативной памяти;

- понимать основы работы с потоками данных.

- понимать принципы повторного использования кода;

- знать способы передачи данных между функциями;

- понимать назначение функций обратного вызова.

- принцы написания безопасного кода;

- директивы препроцессора;

- понимать принципы много поточности.

- знать, как подключить сторонние библиотеки к проекту;

*Уметь:*

- проводить отладку разрабатываемой программы;

- проводить преобразование типов данных;

- создавать программу на базе пустого проекта в VisualStudio.

- выделять и освобождать память из «кучи»;

- записывать данные в файл и читать данные из файла;

- создавать функции и прототипы функций.

- разбивать код на заголовочные файлы и файлы исходного кода;

- функционально делить код.

- писать макросы;

- создавать графический интерфейс;

- использовать библиотеку STL;

- создавать и запускать потоки.

- использовать WinAPIдля передачи данных по сети;

- использовать WinAPIдля работы с файловой системой;

- работать со статическим и динамическим библиотеками;

- работать с базой данным MySql с помощью MySql.dll.

Текущий контроль осуществляется с помощью самостоятельных и практических работ.

Тематический контроль осуществляется по завершении темы в форме контрольной практической работы.

**Формы подведения итогов реализации программы**: урок-презентация работ. Презентация работ выполненных учащимися самостоятельно по результатам курса. Тему работы учащийся выбирает самостоятельно.

**Учебно-тематический план**

|  |  |  |
| --- | --- | --- |
| **№** | **Название темы** | **Количество часов** |
| 1 | Простая программа | 9 |
| 2 | Хранение данных | 30 |
| 3 | Повторное использование кода | 30 |
| 4 | Разработка программы | 44 |
| 5 | Расширение возможностей | 40 |
| 6 | Внешние библиотеки | 31 |
| 7 | Презентация работ | 20 |
|  | **Итого:** | 204 |

**Содержание программы**

**Простая программа – 9 часов**

Введение. Применение языков С/С++. Знакомство со средой разработки. Знакомство с синтаксисом языка С++. Процедурный подход к программированию.

**Практические и самостоятельные работы:**

1. Создание простейшей программы в среде разработки VisualStudio. С применением принципа последовательности.
2. Написание программ использующих циклы: с предусловием, с постусловием, с заданным количеством повторений.
3. Написание программы использующей циклы и операторы ветвления.

**Требования к подготовке учащихся:**

*Учащиеся должны знать/понимать:*

- синтаксис основных операторов языка;

- типы данных;

- понимать принципы преобразования типов данных.

*Уметь:*

- проводить отладку разрабатываемой программы;

- проводить преобразование типов данных;

- создавать программу на базе пустого проекта в VisualStudio.

**Хранение данных – 30 часов**

Указатели. Адреса. Пользовательские типы данных. Разделяемая память. Работа с файлами.

**Практические и самостоятельные работы:**

1. Написание программы использующей указатели и динамическое выделение памяти.
2. Написание программ использующих многомерные статические и динамические массивы.
3. Написание программы генерации случайных строк.
4. Написание программы классификатора с использованием структур (предмет классификации выбирается учащимся).
5. Написание программы считывающей данные из файла и производящую запись в файл.
6. Доработка программы классификатора с применением списков.
7. Доработка программы классификатора с применениембинарных деревьев.

**Требования к подготовке учащихся:**

*Учащиеся должны знать/понимать:*

- понимать принципы выделения памяти из «кучи»;

- понимать способы хранения и адресации данных в оперативной памяти;

- понимать основы работы с потоками данных.

*Уметь:*

- выделять и освобождать память из «кучи»;

- записывать данные в файл и читать данные из файла;

**Повторное использование кода – 30 часов**

Принципы повторного использования кода. Создание функций. Передача данных между функций.

**Практические и самостоятельные работы:**

1. Написание программы генерирующей вещественные случайные числа.
2. Написание программ,разделенной на простые функции и использующей глобальные переменные для передачи данных между функций.
3. Доработка программы из урока 2, отказ от глобальных переменных.
4. Доработка программы по записи в файл. Запись в файл бинарных данных.
5. Программа генерации простых предложений.
6. Доработка программы чтения данных из файла. Чтение бинарных данных.
7. Программа рекурсивного анализа текстового файла. Удаление выделенных фрагментов.
8. Написание программы использующей функции обратного вызова для анализа переданных в нее данных.

**Требования к подготовке учащихся:**

*Учащиеся должны знать/понимать:*

- понимать принципы повторного использования кода;

- знать способы передачи данных между функциями;

- понимать назначение функций обратного вызова.

*Уметь:*

- создавать функции и прототипы функций.

**Разработка программы – 44 часа**

Разбиение кода на заголовочные фалы и файлы исходного кода. Разработка программы методом черного ящика. Безопасный код. Форматирование код и именование переменных.

**Практические и самостоятельные работы:**

1. Написание программы с пользовательским интерфейсов в виде числового меню.

**Требования к подготовке учащихся:**

*Учащиеся должны знать/понимать:*

- принцы написания безопасного кода;

*Уметь:*

- разбивать код на заголовочные файлы и файлы исходного кода;

- функционально делить код.

**Расширение возможностей – 40 часов**

Препроцессор. Макрос. Пространство имен. Библиотека шаблонов. Работа с несколькими потоками. Синхронизация и доступ к общим данным.

**Практические и самостоятельные работы:**

1. Написание программы использующей макросы.
2. Написание программы с пользовательским графическим интерфейсом, с применением WinAPI.
3. Написание программы с несколькими пользовательскими пространствами имен.
4. Написание программ использующих string, vector, map, list.
5. Написание серверного приложения для работы с несколькими клиентами.

**Требования к подготовке учащихся:**

*Учащиеся должны знать/понимать:*

- директивы препроцессора;

- понимать принципы много поточности.

*Уметь:*

- писать макросы;

- создавать графический интерфейс;

- использовать библиотеку STL;

- создавать и запускать потоки.

**Внешние библиотеки – 31 часов**

Подключение внешних библиотек. Настройка зависимостей. Знакомство с WinAPI. Работа с сетью посредствам WinSok. Работа с 2Dи 3D графикой.

**Практические и самостоятельные работы:**

1. Написание программы использующей Unrar.dll.
2. Написание программы использующей MySql.dll.
3. Работа с файловой системой через WinAPI.
4. Написание программыиспользующуюOpenGL для от рисовки графических примитивов.
5. Написание программы использующуюOpenGL для от рисовки 3Dобъекстов.

**Требования к подготовке учащихся:**

*Учащиеся должны знать/понимать:*

- знать, как подключить сторонние библиотеки к проекту;

*Уметь:*

- использовать WinAPIдля передачи данных по сети;

- использовать WinAPIдля работы с файловой системой;

- работать со статическим и динамическим библиотеками;

- работать с базой данным MySql с помощью MySql.dll.

**Презентация работ – 20 часов**

Презентация работ выполненных учащимися самостоятельно. По результатам курса. Тему работы учащийся выбирает самостоятельно.

**Календарно-тематическое планирование**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Календарно-тематический план** | | | | |
| **№ урока** | **Тема урока** | **Элементы содержания** | **Формы контроля** | **Дата проведения** |
| **Простая программа 9 часов** | | | | |
| 1-2 | Структура простой программы | Знакомство со средой разработки VisualStudio. Структура программы на языке С/С++. Типы данных. Стандартные потоки ввода/вывода. | С/р |  |
| 3-4 | Операторы цикла | Операторы цикла: for, while, do…while. | С/р |  |
| 5-9 | Операторы ветвления и безусловные переходы | Оператор ветвления if…else…Конструкции if… elseif… Оператор множественного выбора switch… Оператор безусловного перехода goto… | С/р |  |
| **Хранение данных 30 часов** | | | | |
| 10-13 | Хранение данных в памяти | Представление хранения данных в оперативной памяти компьютера. Классы памяти. Модификаторcons. |  |  |
| 14-17 | Указатели | Переменные указатели и их назначение. Создание указателей. Получение адреса переменной. Выделение памяти из «кучи». | С/р |  |
| 18-21 | Массивы | Статически и динамические массивы. Многомерные массивы. Выделение памяти под массив из «кучи». | С/р |  |
| 22-25 | Работа со строками | Работа со строками как с последовательностями charи wchar\_t. | С/р |  |
| 26- 29 | Структуры | Структуры назначение создание. Объединения. | С/р |  |
| 30-33 | Работа с файлами | Файловый ввод/вывод посредством объектов ifstreame/ofstreame | С/р |  |
| 34-36 | Списки | Одно- и дву- направленные списки. Создание списков без использования STL. | Пр/р |  |
| 37-39 | Деревья | Бинарные деревья. Сбалансированные бинарные деревья обход дерева. Создание деревьев без использования STL. | Пр/р |  |
| **Повторное использование кода 30 часов** | | | | |
| 40-43 | Стандартные библиотеки С/С++ | Знакомство со стандартными библиотеками math, timeи проч. | С/р |  |
| 44-47 | Простые функции | Простые функции void … ()  Создание функций.  Создание прототипов функций.  Область видимости переменных.  Модификатор static.  Глобальные переменные. | С/р |  |
| 48-51 | Аргументы функций | Функции, принимающие аргументы.  Создание функций и прототипов функций, принимающих аргументы.  Перегрузка. | С/р |  |
| 52-55 | Типы функций | Функции, возвращающие значения.  Создание функций и прототипов функций,возвращающих значения. | С/р |  |
| 56-59 | Возврат значений функцией | Использование ссылок для передачи аргументов в функцию. | Пр/р |  |
| 60-63 | Обмен данными между функциями | Использование указателей для передачи данных меду функциями. | Пр/р |  |
| 64-66 | Рекурсия | Понятие рекурсии. Рекурсивные функции. | С/р |  |
| 67-69 | Функции обратного вызова | Назначение функций обратного вызова. Создание функций обратного вызова. | С/р |  |
| **Разработка программы 44 часа** | | | | |
| 70- 80 | Многофайловая программа | Создание многофайловых программ. Файлы исходного кода. Заголовочные файлы. Модификатор external. |  |  |
| 81-91 | Метод черного ящика | Разработка программ методом черного ящика. |  |  |
| 92-102 | Создание пользовательского интерфейса | Создание пользовательского интерфейса для консольного приложения в виде цифрового меню. | С/р |  |
| 103-113 | Легко читаемы код, безопасный код | Общие правила именования переменных и классов. Форматирование кода. Правильный подход к созданию безопасного кода. |  |  |
| **Расширение возможностей 40 часов** | | | | |
| 114-121 | Препроцессор | Директивы препроцессора и макросы. Создание макросов. Знакомство с директивами. | С/р |  |
| 122-129 | Создание пользовательского графического интерфейса | Создание пользовательского графического интерфейса с использованием WinAPI. | С/р |  |
| 130-137 | Пространство имен | Понятие пространства имен. Использование пространства имен. Разбиение программы на пространства имен. | С/р |  |
| 138-145 | Библиотека шаблонов STL | Знакомство с библиотекой STL. String, vector, List, Map. | С/р |  |
| 146-153 | Многопоточность | Принципы построения многопоточных программ. Синхронизация. Создание потоков. Мьютексы, симфоры, критические секции. | Пр/р |  |
| **Внешние библиотеки 31 часов** | | | | |
| 154-159 | Статические и динамические библиотеки | Назначение статических и диадических библиотек. Подключение библиотек к своей программе. Настройка зависимостей. |  |  |
| 160-165 | Работа с динамической библиотекой на примере Unrar.dll | Практическая работа с динамической библиотекой Unrar.dll. | Пр/р |  |
| 166-171 | Работа с динамической библиотекой на примере MySql.dll | Практическая работа с динамической библиотекой MySql.dll. | Пр/р |  |
| 172-177 | WinAPI | Работа с файловой системой через WinAPI. Работа с сетью через WinAPI.Создание клиентского и серверного приложений, для обмена данными через сеть. | Пр/р |  |
| 178-184 | OpenGL 2D графика | Знакомство с OpenGL работа с двумерными изображениями. | Пр/р |  |
| **Презентация работ 20 часов** | | | | |
| 185-204 | Презентация практических работ | Презентация и обсуждение работ. | Пр/р |  |

**Методическое обеспечение программы**

**Перечень средств ИКТ, необходимых для реализации программы**

Аппаратные и программные средства:

* Компьютеры
* Проектор
* ОС Windows
* Программы:Visual Studio.