

**Министерство просвещения Российской Федерации
Министерство образования Иркутской области
Департамент образования города Иркутска
Муниципальное автономное общеобразовательное учреждение Лицей ИГУ города Иркутска
МАОУ Лицей ИГУ г. Иркутска**

РАССМОТРЕНО

на заседании методического объединения
учителей информатики и технологии от
29.08.2023г. протокол №1.
Руководитель МО Л.Н. Шеметова

УТВЕРЖДЕНО

Приказ № 01-06-140 от
30.08.2023 г.
Директор Е.Ю. Кузьмина

ПРИНЯТО

решением педагогического совета
от 30.08.2023 г., протокол №1

ID -

РАБОЧАЯ ПРОГРАММА

**ID ---
учебного курса**

**УЧЕБНОГО КУРСА «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++, РЕШЕНИЕ ЗАДАЧ
ПОВЫШЕННОЙ СЛОЖНОСТИ И ОЛИМПИАДНЫХ ЗАДАЧ»**

(для учащихся 9-го классов)

Срок освоения – 1 год

Уровень сложности программы углубленный

Количество часов по программе 34 (17)

Составитель программы: Семеней Е.И учитель информатики, высшая кв.категория

г.Иркутск, 2023 г.

АННОТАЦИЯ К РАБОЧЕЙ ПРОГРАММЕ СПЕЦКУРСА
«Программирование на языке С++, решение задач повышенной сложности»
для 9-го класса

Рабочая программа спецкурса «Программирование на языке С++, решение задач повышенной сложности» (9 класс) разработана в соответствии с требованиями ФГОС и ФОП основного общего образования и Положением «О рабочих программах учебных предметов, учебных курсов (в том числе внеурочной деятельности), учебных модулей в соответствии с требованиями ФГОС и ФОП основного общего образования» МАОУ Лицея ИГУ г.Иркутска, утвержденного приказом директора 01-06-132 от 30.08.2023 года и является частью основной образовательной программы основного общего образования.

Рабочая программа ориентирована на целевые приоритеты, сформулированные в федеральной рабочей программе воспитания и в рабочей программе воспитания МАОУ Лицей ИГУ г. Иркутска.

Обучение курса направлено на достижение следующих целей:

- формирование алгоритмической культуры обучающихся, развитие их алгоритмического мышления;
- обучение структурному программированию как методу, предусматривающему создание понятных, простых и удобочитаемых программ, характерными особенностями которых являются модульность, использование унифицированных структур следования, выбора и повторения, отказ от неструктурированных передач управления, ограниченное использование глобальных переменных;
- приобретение обучающимися знаний и навыков алгоритмизации в ее структурном варианте;
- освоение обучающимися всевозможных методов решения задач, реализуемых на языке СИ++;
- формирование у обучающихся навыков грамотной разработки программы;
- углубление знаний, умений и навыков решения задач по программированию и алгоритмизации.
- формирование навыков самостоятельной исследовательской работы;

В результате изучения программы спецкурса предполагается дальнейшее развитие личностных, регулятивных, коммуникативных и познавательных универсальных учебных действий, учебной (общей и предметной) и общепользовательской ИКТ-компетентности учащихся.

Рабочая программа учебного курса «Программирование на языке С++, решение задач повышенной сложности» реализуется за счет части, формируемой участниками образовательных отношений, входит в предметную область «Математика и информатика»

Срок реализации программы – 1 год (9 класс)

Количество учебных часов, на которые рассчитана программа

	9 класс (вариант 1)	9 класс (вариант 2)
Кол-во учебных недель	34	17
Кол-во часов в неделю	1	0,5
Кол-во часов в год	34	17

В программу включены содержание, планируемые результаты (личностные, метапредметные, предметные), тематическое планирование с учетом рабочей программы воспитания, оценочные и методические материалы.

Рабочая программа рассмотрена на заседании методического объединения учителей-предметников (протокол №1 от 29.08.2023 г.), согласована с заместителем директора МАОУ Лицей

ИГУ г. Иркутска, утверждена к использованию в образовательной деятельности Лицея приказом директора № 01-06-140 от 30.08.2023 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Компьютеризация общества развивается так стремительно, что в настоящее время охватывает практически все сферы жизнедеятельности человека. Культура общения с компьютером становится частью общей культуры человека, даже, если в дальнейшем он не свяжет свою профессиональную деятельность с программированием, пользователем различных программ он обязательно будет, а для того, чтобы пользоваться сложными инструментами, такими как электронная процессор MS Excel, СУБД MS Access, необходимо иметь навыки алгоритмического мышления. Поэтому возникает потребность в развитии навыков алгоритмического мышления и алгоритмического подхода к решению задач. Программирование – это быстрый, конкретный и подходящий способ приобретения навыков алгоритмического мышления.

Актуальность программы спецкурса состоит в том, современные профессии становятся все более интеллект ёмкими, требующими развитого логического мышления. Поэтому для подготовки учащихся к жизни в современном информационном обществе в первую очередь необходимо развивать логическое мышление, способности к анализу и синтезу. Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и некоторые общие мыслительные навыки, способствует формированию научного мировоззрения, стиля жизни современного человека. Также нельзя не отметить и то, что в настоящее время языки программирования на основе C++ заняли ведущее место в сфере программирования. Для освоения основ программирования на языке C++ , для решения задач повышенной сложности и предназначена данная программа спецкурса. Содержание курса способствует устойчивому и эффективному развитию системного и алгоритмического мышления обучающихся.

Новизна программы связана с возможностью реализации концепции развивающего обучения, так как непременным атрибутом программирования является постоянная оценка своих действий. В процессе работы над программой учащийся имеет возможность осознать, насколько правильно принятое решение, насколько верен ход рассуждений, все ли факты учтены при составлении алгоритма решения задачи. Обзор и использование современных возможностей, подкрепленный постоянной отработкой базовых элементов языка, формирует профессиональное мышление учащихся, склонность к самостоятельному углублению знаний.

Спецкурс «Программирование на языке C++, решение задач повышенной сложности и олимпиадных задач» предназначен для работы с учащимися, проявившими интерес к изучению языков программирования, где они могут получить необходимые знания, найти единомышленников и друзей по интересам, а в дальнейшем, получать удовольствие от решения сложных задач.

Обучение программированию начинается с изучения алгоритмов и основ языка программирования C++, как инструмента реализации алгоритмов.

Программа предусматривает концентрическое изучение материала, что позволяет по мере изучения курса давать всё более глубокие знания по каждому из изучаемых разделов программирования, не теряя при этом целостности изложения всего материала. Такой подход расширяет кругозор учащихся и дает возможность самостоятельно решать нестандартные задачи.

Преподавание разработанного спецкурса построено на побуждении мотивации к изучению предлагаемого материала. Его теоретическая часть сопровождается специально подобранными примерами и упражнениями, выполненными с применением и анализом различных методических и технологических приемов. При изучении каждого раздела учащимся предлагается большое количество задач и упражнений для самостоятельного и коллективного решений с учетом индивидуальных способностей каждого. Нужно отметить, что теоретическая часть курса даётся параллельно с практической работой, это позволяет избежать перегрузки учащихся и сохранить их внимание в течение занятия.

Исходя из того, что знания присваиваются человеком только лишь в результате его самостоятельной учебно-познавательной деятельности, занятия строятся следующим образом: учащимся предлагается задача для самостоятельного решения, затем проводится коллективный разбор проблем, возникших при ее решении, и способов их преодоления. Главным на данном этапе является создание атмосферы коллективного творчества, заинтересованного и квалифицированного обсуждения возникающих вопросов. В случае необходимости коллективно решаются еще одна или несколько задач на данную тему. Здесь важно дать возможность учащимся показать различные методы решения, обсудить возникшие сложности и ошибки. В том случае, если это необходимо, в начале, либо в ходе занятия возможны теоретические отступления, позволяющие связать изучаемый материал с ранее усвоенными знаниями.

Такое построение спецкурса позволяет придать ученику уверенность в своих силах, побудить его к самостоятельному исследованию возникающих вопросов, развить способность к творчеству, конструктивной коллективной работе.

Основной целью курса является формирование алгоритмического мышления и обучение искусству программирования.

Задачи курса

- развитие логико-алгоритмического, пооперационного и системного мышления учащихся
- формирование у обучающихся навыков грамотной разработки программы;
- углубление знаний, умений и навыков решения задач по программированию и алгоритмизации;
- формирование умений использования компьютерного программирования для решения практических задач из различных предметных областей;
- формирование навыков самостоятельной исследовательской работы
- подготовка учащихся к современной деятельности для решения различных практических задач;
- формирование знаний учащихся в области теории информации, основ алгоритмизации и программирования;
- подготовка учащихся в области основных содержательных компонентов информационной культуры: технологического, коммуникативного, алгоритмического;
- формирование и развитие интереса учащихся к компьютерным технологиям и методам обработки информации.

Планируемые результаты образовательной деятельности при реализации программы

В результате изучения спецкурса получают дальнейшее развитие личностные, регулятивные, коммуникативные и познавательные универсальные учебные действия, учебная (общая и предметная) и общепользовательская ИКТ-компетентность обучающихся.

Личностные

Личностные результаты имеют направленность на решение задач воспитания, развития и социализации обучающихся средствами учебного предмета.

1) **патриотического воспитания**: ценностное отношение к отечественному культурному, историческому и научному наследию, понимание значения информатики как науки в жизни современного общества, владение достоверной информацией о передовых мировых и отечественных достиже-

ниях в области информатики и информационных технологий, заинтересованность в научных знаниях о цифровой трансформации современного общества;

2) **духовно-нравственного воспитания:** ориентация на моральные ценности и нормы в ситуациях нравственного выбора, готовность оценивать своё поведение и поступки, а также поведение и поступки других людей с позиции нравственных и правовых норм с учётом осознания последствий поступков, активное неприятие асоциальных поступков, в том числе в Интернете;

3) **гражданского воспитания:** представление о социальных нормах и правилах межличностных отношений в коллективе, в том числе в социальных сообществах, соблюдение правил безопасности, в том числе навыков безопасного поведения в Интернет-среде, готовность к разнообразной совместной деятельности при выполнении учебных, познавательных задач, создании учебных проектов, стремление к взаимопониманию и взаимопомощи в процессе этой учебной деятельности, готовность оценивать своё поведение и поступки своих товарищей с позиции нравственных и правовых норм с учётом осознания последствий поступков;

4) **ценностей научного познания:** сформированность мировоззренческих представлений об информации, информационных процессах и информационных технологиях, соответствующих современному уровню развития науки и общественной практики и составляющих базовую основу для понимания сущности научной картины мира; интерес к обучению и познанию, любознательность, готовность и способность к самообразованию, осознанному выбору направленности и уровня обучения в дальнейшем;

овладение основными навыками исследовательской деятельности, установка на осмысление опыта, наблюдений, поступков и стремление совершенствовать пути достижения индивидуального и коллективного благополучия;

сформированность информационной культуры, в том числе навыков самостоятельной работы с учебными текстами, справочной литературой, разнообразными средствами информационных технологий, а также умения самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

5) формирования культуры здоровья:

осознание ценности жизни, ответственное отношение к своему здоровью, установка на здоровый образ жизни, в том числе и за счёт освоения и соблюдения требований безопасной эксплуатации средств информационных и коммуникационных технологий;

6) трудового воспитания:

интерес к практическому изучению профессий и труда в сферах профессиональной деятельности, связанных с информатикой, программированием и информационными технологиями, основанными на достижениях науки информатики и научно-технического прогресса;

осознанный выбор и построение индивидуальной траектории образования и жизненных планов с учётом личных и общественных интересов и потребностей;

7) экологического воспитания:

осознание глобального характера экологических проблем и путей их решения, в том числе с учётом возможностей информационных и коммуникационных технологий;

8) адаптации к изменяющимся условиям социальной среды:

освоение обучающимися социального опыта, основных социальных ролей, соответствующих деятельности возраста, норм и правил общественного поведения, форм социальной жизни в группах и сообществах, в том числе существующих в виртуальном пространстве.

Метапредметные результаты,

- умение самостоятельно планировать пути достижения целей, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;

- умения соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;

- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;

- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (далее ИКТ-компетентности);

- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение; формирование и развитие компетентности в области использования информационно-коммуникационных технологий (далее ИКТ-компетенции).

Регулятивные универсальные учебные действия

Обучающийся научится:

- целеполаганию, включая постановку новых целей, преобразование практической задачи в познавательную;

- самостоятельно анализировать условия достижения цели на основе учета выделенных учителем ориентиров действия в новом учебном материале;

- планировать пути достижения целей;

- уметь самостоятельно контролировать свое время и управлять им.

Коммуникативные универсальные учебные действия

Обучающийся научится:

- учитывать разные мнения и стремиться к координации различных позиций в сотрудничестве;

- аргументировать свою точку зрения, спорить и отстаивать свою позицию не враждебным для оппонентов образом;

- задавать вопросы, необходимые для организации собственной деятельности и сотрудничества с партнером;

- осуществлять взаимный контроль и оказывать в сотрудничестве необходимую взаимопомощь;

- устанавливать и сравнивать разные точки зрения, прежде чем принимать решения и делать выбор;

Познавательные универсальные учебные действия

Обучающийся научится:

- осуществлять выбор наиболее эффективных способов решения задач, в зависимости от конкретных условий;

- давать определение понятиям;

- обобщать понятия — осуществлять логическую операцию перехода от видовых признаков к родовому понятию, от понятия с меньшим объемом к понятию с большим объемом;

- устанавливать причинно-следственные связи;

- создавать и преобразовывать модели и схемы для решения задач.

Предметные результаты

- умение использовать термины понятий «алгоритм», «данные», «программа» через призму практического опыта в ходе создания программных кодов; понимание различий между употреблением этих терминов в обыденной речи и в информатике;

- умение формально выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов;

- умение создавать и выполнять программы для решения задач различного уровня сложности на языке программирования C++;

СОДЕРЖАНИЕ ПРОГРАММЫ

Содержание разделов спецкурса	34 часа	17 часов	Электронные ресурсы
<p><u>I раздел.</u> Основные элементы языка программирования C++</p> <p>В данном разделе рассматривается понятие линейного алгоритма.</p> <p>Основное место уделяется изучению операторов языка программирования C++:</p> <ul style="list-style-type: none"> - оператор присваивания; - составной оператор, который служит для записи в свое тело других операторов, последовательность которых рассматривается в данном случае как один оператор; - операторы ввода/вывода данных <p>Учащиеся научатся создавать, вводить в компьютер, выполнять и исправлять простейшие программы на языке C++, закрепят и отработают навыки решения задач с использованием линейного программирования на языке C++.</p>	3 час.	2 час.	
<p><u>II раздел.</u> Операторы языка C++</p> <p>В данном разделе рассматривается понятие алгоритма с ветвлением.</p> <p>Учащиеся смогут обучиться программированию алгоритмов с ветвлением на языке C++, подробно изучат оператор условного выполнения.</p> <p>Кроме оператора условного выполнения в C++ имеется ещё одна управляющая конструкция, одно из названий которой - оператор выбора switch.</p> <p>На самом деле это усложнённый оператор if, он позволяет программе выполняться не двумя способами, в зависимости от выполнения условия, а несколькими, в зависимости от значения некоторого выражения.</p> <p>Учащиеся научатся создавать, вводить в компьютер, выполнять и исправлять программы на языке C++, закрепят и отработают навыки решения задач с использованием алгоритмов с ветвлением на языке C++.</p>	4 час.	2 час.	

<p><u>III раздел. Циклические конструкции</u></p> <p>В данном разделе рассматривается понятие циклического алгоритма.</p> <p>Изучаются основные разновидности циклов:</p> <ul style="list-style-type: none"> - конструкция for - конструкция while - конструкция do/while - вложенные циклы <p>Учащиеся познакомятся с правилами, которые необходимо соблюдать при использовании в программе оператора цикла.</p> <p>Учащиеся научатся создавать, вводить в компьютер, выполнять и редактировать программы на языке C++, закрепят и отработают навыки решения задач с использованием циклических алгоритмов на языке C++.</p>	11 час.	5 час.	
<p><u>IV раздел. Массивы</u></p> <p>В данном разделе учащиеся познакомятся с составным типом данных – массивами, узнают способы описания одномерных и двумерных массивов, способы обращения к элементам одномерных и двумерных массивов, узнают способы формирования одномерных и двумерных массивов.</p> <p>Учащиеся научатся писать и отлаживать программы формирования, ввода, вывода одномерных и двумерных массивов, проводить поиск элементов в двумерном массиве по разным условиям, проводить логический анализ программ, их тестирование.</p>	8 час.	4 час.	
<p><u>V раздел. Функции как программные модули C++</u></p> <p>Изучение данного раздела поможет узнать: Почему наличие полноценных функций является принципиально важным для структурно-ориентированного языка высокого уровня. Каковы правила описания функций в C++. В чем принципиальные отличия между формальными, локальными и глобальными переменными. В чем отличия между параметрами-переменными и параметрами-значениями, и в каких ситуациях целесообразно использовать те и другие. В чем в</p>	5 час.	2 час.	

принципе заключается рекурсия и как она реализована на C++. Выделять вспомогательные алгоритмы в сложных задачах. Формировать функции. Правильно строить обращения к функциям.			
VI раздел. Функции и объекты класса string В данном разделе учащиеся познакомятся с понятием строки, строковым типом данных, узнают способы описания строк, научатся пользоваться операциями и функциями над строками, решать задачи по обработке строк.	3 час.	2 час.	

ВАРИАНТ 1

Тематическое планирование программы спецкурса «Программирование на языке языке C++, решение задач повышенной сложности и олимпиадных задач» (34 часа)

№	Но-мер урока	Тема урока	Цели	Форма занятий
1	Основные элементы языка программирования C++ (3 часа)			
	1.	Алфавит языка. Структура программы. Простые типы данных. Переменная.		Лекция
	2.	Оператор присваивания. Арифметические выражения. Ввод, вывод данных. Стандартные функции.		Лекция и практическая работа
	3.	Среда программирования C++. Программирование линейных алгоритмов.		Лекция и практическая работа
2	Операторы, для реализации алгоритмов с ветвлением (4 часа)			
	4.	Выбор действий в алгоритмах. Формы ветвления. Условный оператор. Составной оператор.		Лекция
	5.	Программирование разветвляющихся алгоритмов.		Практическая работа
	6.	Сложные логические выражения. Логические операции. Программирование разветвляющихся алгоритмов.		Лекция и практическая работа
	7.	Решение задач по теме "Программирование разветвляющихся алгоритмов".		Практическая работа
3	Циклические конструкции (11 часов)			
	8.	Алгоритмы, содержащие циклические структуры. Оператор цикла с предусловием.		Лекция
	9.	Разбор задач по теме: «Цикл с предусловием».		Практическая работа
	10.	Цикл с предусловием: написание и отладка программ.		Практическая работа
	11.	Оператор цикла с постусловием.		Лекция
	12.	Разбор задач по теме: «Цикл с постусловием».		Практическая работа
	13.	Цикл с постусловием: Написание и отладка программ.		Практическая работа

	14.	Оператор цикла с параметром.	Лекция
	15.	Разбор задач по теме: «Цикл с параметром».	Практическая работа
	16.	Цикл с параметром: Написание и отладка программ.	Практическая работа
	17.	Структура вложенных циклов.	Лекция
	18.	Написание и отладка программ, реализующих циклические алгоритмы.	Практическая работа
4	Массивы (8 часов)		
	19.	Нестандартные типы данных. Тип – массивы в C++. Одномерные массивы..	Лекция
	20.	Генератор случайных чисел. Поиск в массиве.	Лекция и практическая работа
	21.	Методы работы с элементами одномерного массива.	Лекция
	22.	Написание и отладка программ.	Практическая работа
	23.	Двумерные массивы.	Лекция
	24.	Обработка элементов двумерного массива.	Лекция и практическая работа
	25.	Написание и отладка программ обработки двумерных массивов.	Практическая работа
	26.	Решение задач по теме "Обработка массивов". Написание и отладка программ по обработке элементов массивов.	Практическая работа
5	Функции как программные модули C++ (5 часов)		
	27.	Функции в языке C++ : объявление, определение	Лекция
	28.	Передача параметров по значению. Написание и отладка программ.	Лекция и практическая работа
	29.	Передача параметров по указателю. Возвращаемые значения.	Практическая работа
	30.	Написание и отладка программ с использованием функций пользователя.	Практическая работа
	31.	Решение задач по теме «Функции пользователя».	Практическая работа
6	Функции и объекты класса string (3 часа)		
	32.	Обработка символьных данных. Операции над символьными данными. Стандартные функции, используемые для обработки символьных данных.	Лекция
	33.	Тип данных String. Стандартные процедуры и функции для обработки строковых данных.	Лекция и практическая работа
	34.	Решение задач по теме "Обработка строковых данных".	Практическая работа

ВАРИАНТ 2**Тематическое планирование программы спецкурса «Программирование на языке языке C++, решение задач повышенной сложности и олимпиадных задач» (17 часов)**

№	Но-мер урока	Тема урока	Цели	Форма занятий
1	Основные элементы языка программирования C++ (2 часа)			
	1.	Алфавит языка. Структура программы. Простые типы данных. Переменная.		Лекция
	2.	Оператор присваивания. Арифметические выражения. Ввод, вывод данных. Стандартные функции. Среда программирования C++. Программирование линейных алгоритмов		Лекция и практическая работа
2	Операторы, для реализации алгоритмов с ветвлением (2 часа)			
	3.	Выбор действий в алгоритмах. Формы ветвления. Условный оператор. Составной оператор.		Лекция и практическая работа
	4.	Сложные логические выражения. Логические операции. Программирование разветвляющихся алгоритмов.		Лекция и практическая работа
3	Циклические конструкции (5 часов)			
	5.	Алгоритмы, содержащие циклические структуры. Оператор цикла с предусловием и постусловием.		Лекция
	6.	Разбор задач по теме: «Цикл с предусловием и аостусловием».		Практическая работа
	7.	Оператор цикла с параметром.		Лекция
	8.	Разбор задач по теме: «Цикл с параметром».		Практическая работа
	9.	Структура вложенных циклов.		Лекция и практическая работа
4	Массивы (4 часа)			
	10.	Нестандартные типы данных. Тип – массивы в C++. Одномерные массивы. Генератор случайных чисел. Поиск в массиве.		Лекция и практическая работа
	11.	Методы работы с элементами одномерного массива. Написание и отладка программ.		Лекция и практическая работа
	12.	Двумерные массивы. Обработка элементов двумерного массива.		Лекция
	13.	Решение задач по теме "Обработка массивов". Написание и отладка программ по обработке элементов массивов.		Практическая работа
5	Функции как программные модули C++ (2 часа)			
	14.	Функции в языке C++ : объявление, определение. Передача параметров по значению. Написание и отладка программ. Передача параметров по указателю. Возвращаемые значения.		Лекция
	15.	Написание и отладка программ с использованием функций пользователя.		Практическая работа
6	Функции и объекты класса string (2 часа)			

	16.	Обработка символьных данных. Операции над символьными данными. Стандартные функции, используемые для обработки символьных данных.	Лекция
	17.	Тип данных String. Стандартные процедуры и функции для обработки строковых данных.	Лекция и практическая работа

УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

ОБЯЗАТЕЛЬНЫЕ УЧЕБНЫЕ МАТЕРИАЛЫ ДЛЯ УЧЕНИКА

1. Информатика 10-11 классы. Программирование на языке С++(материалы к занятиям)/ С.В.Феськов. – Волгоград: Учитель, 2009.
2. Беляев С.Н., Лалетин Н.В. Региональные олимпиады по информатике – 2008/2009: учебно-методическое пособие; Краснояр. гос. пед. ун-т. В.П. Астафьева. – Красноярск, 2009.

Учебные пособия для учащихся

1. Культин Н.Б. С/С++ в задачах и примерах. – СПб.: БХВПетербург, 2006, - 288 с.
2. Стивен Прата Язык программирования С++. Лекции и упражнения. – М.: ООО «И.Д. Вильямс», 2012.
3. Березин Б.И., Березин С.Б. Начальный курс С и С++. - М.: ДИАЛОГ_МИФИ, 2005. - 288 с.

Учебные пособия для учителя

1. Касаткин А.И., Вальвачев А.Н. Профессиональное программирование на языке Си от Turbo С к Borland С++; Справ. Пособие под общей ред. А.И.Касаткина, Мн.: Высш. Школа, 1992, 240 с.
2. Франка П. С++: учебный курс. - СПб.: Питер, 2001. - 528 с.: ил.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. / Т.А. Павловская. - СПб.: Питер, 2002. - 464 с.: ил.
4. А.И. Гусева Учимся информатике: задачи и методы их решения. – М. Диалог МИФИ, 1998.
5. Бруко Бабэ. Просто и ясно о Borland С++; Пер. с англ. — М.: БИНОМ. 1997г. — 416с.
6. Страуструп Б. Язык программирования Си++; Пер. с англ. — М.: Радио и связь, 1991г. — 352с.: ил.
7. Уэйт М., Прата С. Язык Си. Руководство для начинающих. Пер. с англ. — М.: Мир, 1988г.512с.
8. Хэнкок Л., Кригер М. Введение в программирование на языке Си. — М.: Радио и связь, 1986г. — 192с.
9. Акатнова, С. В. Алексахин и др. – М. : Просвещение, 2002.
10. Д.М. Златопольский Я иду на урок информатики: Задачи по программированию. 7-11 кл. М. Первое сентября, 2001 г
11. Житкова О.А., Кудрявцева Е.К. Алгоритмы и основы программирования. М.: «Интеллект-центр», 2001.
12. Киммел П. и др. Borland С++ 5: пер. с англ. - Санкт-Петербург, 1997.- 976 с., ил.

ЦОР (цифровые образовательные ресурсы: диски, Интернет-источники)

1. <http://informatics.mccme.ru/course/view.php?id=666>
2. <http://kpolyakov.spb.ru/school/probook/tests.htm>
3. [http://acmp.ru/catalog/;](http://acmp.ru/catalog/)
4. [http://www.openclass.ru/;](http://www.openclass.ru/)
5. [http://www.uchportal.ru/;](http://www.uchportal.ru/)
6. [http://festival.1september.ru/;](http://festival.1september.ru/)
7. <http://klyaksa.net> и др.

**Приложения к программе спецкурса:
«Программирование на языке C++,
решение задач повышенной сложности и олимпиадных задач»
для учащихся 9-го класса**

1. Формы организации учебных занятий.

- лекционные занятия;

- уроки - практикумы;

В содержании программы спецкурса 2 части:

- теоретическая;
- практическая.

В теоретической части рассматриваются основные понятия языка программирования C++, основные алгоритмические конструкции.

В практической части предлагаются практические работы, направленные на отработку основных алгоритмических конструкций, на развитие логического мышления, на реализацию математических способностей учащихся в ходе составления программ.

Система оценки результатов

Системой оценки результатов освоения программы спецкурса является уровень сформированных знаний, умений, навыков, уровень развития учащихся, включающий индивидуальные качества и личностный рост.

Качество программ, написанных учащимися оценивается следующими критериями:

- Последовательность действий при разработке программ: постановка задачи, выбор метода решения, составление алгоритма, составление программы, запись программы в компьютер, отладка программы, тестирование программы.
- «Правила хорошего тона» при разработке программ: читаемость и корректность программ, защита от неправильного ввода, понятия хорошего и плохого «стиля программирования».

Контроль за усвоением качества знаний должен проводиться на трех уровнях:

1-й уровень – воспроизводящий (репродуктивный) – предполагает воспроизведение знаний и способов деятельности. Учащийся воспроизводит учебную информацию, выполняет задания по образцу.

2-й уровень – конструктивный предполагает преобразование имеющихся знаний. Учащийся может переносить знания в измененную ситуацию, в которой он видит элементы, аналогичные усвоенным;

3-й уровень – творческий предполагает овладение приемами и способами действия. Учащийся осуществляет перенос знаний в незнакомую ситуацию, создает новые нестандартные алгоритмы познавательной деятельности.

1.1. Формы контроля

Текущий контроль знаний осуществляется по результатам выполнения учащимися практических заданий.

Итоговый контроль: Используются следующие методы отслеживания результативности: педагогический анализ результатов тестирования, участие учащихся в лицейских, муниципальных и региональных конкурсах и олимпиадах по программированию, защита проектов по созданию собственных программных продуктов.

В программе спецкурса, рассчитанной на 34 часа (1 час в неделю), не изучаются такие темы, как оператор множественного выбора, рекурсия, рекурсивные функции. Не предусмотрено решение олимпиадных задач.

Оценочные материалы

Приложение 1

Тест по теме: «Операторы ввода-вывода, операторы ветвления»

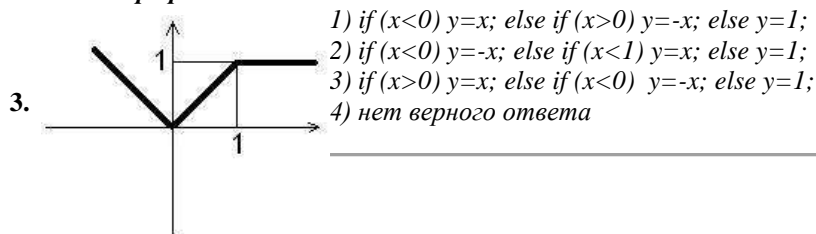
1. Какое значение будет иметь переменная z при $x=1, y=-1$ после выполнения операторов:

```
z=0;
if (x>0)
    if (y>0) z=1; else z=2;
```

Какое значение будет иметь переменная z при $x=-1, y=1$ после выполнения операторов:

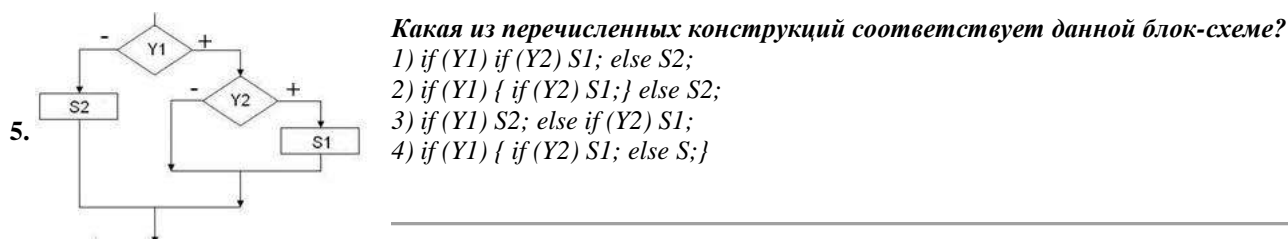
2. $z=0;$
 $if (x>0) if (y>0) z=1; else z=2;$

С помощью какого оператора (из числа приведённых ниже) можно определить значение функции, представленной графически?



Выберите фрагмент программы, в котором переменной y присваивается значение $|x|$, если $-1 < x < 1$, и 1 в противном случае:

4. 1) $if (x>-1) if (x<1) y=abs(x); y=1;$
 2) $y=1; if (x>-1) if (x>1) y=abs(x);$
 3) $if (abs(x)<1) y=abs(x); else y=1;$
 4) $if (x<-1) if (x>1) y=1; y=abs(x);$



Какие из нижеперечисленных строк не содержат синтаксических ошибок?

6. 1) $if (x<y) x=0; else y=0;$
 2) $if (x>y) x=0; else cin>>y;$
 3) $if (x>=y) x=0; y=0; else cout<<z;$
 4) $if (a<b<c) z=z+1;$
 5) $if (sqrt(z)<3,17) z=z+1;$

7. Как называются две формы условного оператора: $if (A) B; else C;$ и $if (A) B;$
 1) длинная и короткая

- 2) полная и неполная
- 3) структурированная и простая
- 4) иерархическая и линейная

Какие из следующих записей содержат ошибки в записи оператора ввода:

- 8.
- 1) `cin >> x >> y >> z;`
 - 2) `cin >> x, y, z;`
 - 3) `cin >> x;`
 - 4) `x = cin >> x;`
 - 5) `cin >> a; b;`
 - 6) `cin >> a >> b + c;`
-

Какие из следующих последовательностей содержат ошибки в записи операторов вывода:

- 9.
- 1) `cout << x << y;`
 - 2) `cout << b, c;`
 - 3) `cout << 100;`
 - 4) `cout << x + 1 << y + 2;`
-

10. Укажите условие выбора чисел, кратных 3, но не кратных 6:

- 1) `(X % 3 == 0);`
- 2) `(X % 3 == 0 && X % 6 != 0);`
- 3) `(X % 3 == 0 || X % 6 != 0);`

11. `switch (k)`

{

}

Здесь `k` не может быть типа:

- 1) `int`
- 2) `char`
- 3) `double`
- 4) `bool`

12. Что неправильно в операторе `switch`?

```
double d;  
d = a + c * 0.2;  
switch (d)  
{  
    case 2: a = b; break;  
    case 4: a = c; break;  
}
```

- 1) после `switch` должно стоять двоеточие;
- 2) переменная `d` имеет недопустимый для `switch` тип;
- 3) нет ветви `default`.
- 4) в строках 5 и 6 нет операторных скобок;

Тест по теме «Двумерные массивы».

1. Укажите правильное описание двумерного массива:

- A) `double a: array [1...n, 1...n];`
- Б) `integer a: array [n][m];`
- В) `int a[n][m];`
- Г) `integer a [n][m];`

2. Что определяет для массива `X[n][m]` следующий алгоритм

`s = 0;`

```

for(i=1; i<= n; i++)
  for(j=1; i<= m; j++)
    if (x[i, j ] <0) s=s+x[i, j ];

```

- А) минимальный элемент массива;
- Б) сумму отрицательных элементов массива;
- В) максимальный элемент массива;
- Г) количество отрицательных элементов массива.

3. Что определяет для массива $x[n][m]$ следующий алгоритм

```

for(i=1; i<= n; i++)
  for(j=1; i<= m; j++)
    if (i % 2 == 0) x[i, j] = a;

```

- А) минимальный элемент массива;
- Б) четные столбцы матрицы заменить на a;
- В) нечетные строки матрицы заменить на a;
- Г) количество четных элементов массива;

4. Что определяет для массива $X[m][m]$ следующий алгоритм

```

for i= 1 to m do
  for(j=1; j<= m; j++)
    if ( (i % 2 )!=0) X[j][i] = A;

```

- А) нечетные столбцы матрицы заменить на A;
- Б) четные столбцы матрицы заменить на A;
- В) количество четных элементов массива;
- Г) нечетные строки матрицы заменить на A;

5. Что определяет для массива $X[n][n]$ следующий алгоритм

```

S=0;
for(i=1; i<= n; i++)
  for (j= 1; j<=n; j++)
    if ( i ==(n-j+1) ) S= S+X[i][j];

```

- А) сумма элементов главной диагонали матрицы;
- Б) четные столбцы матрицы заменить на A;
- В) четные строки матрицы заменить на A;
- Г) сумма элементов побочной диагонали матрицы;

6. Что определяет для массивов $X[n][m]$ и $Y[n][m]$ следующий алгоритм

```

for (i=1; i<=n ; i++)
  for (j=1; j<=m; j++)
    X[i][j] = X[i][j] + Y[i][j];

```

- А) сумма элементов главной диагонали матрицы;
- Б) сумма элементов каждой строки матриц X и Y;
- В) четные строки матрицы заменить на A;
- Г) сумма матриц X и Y.

7. Что определяет для массива $X[[n][m]$ следующий алгоритм

```

V= 0;
for (i=1; i<=n ; i++)
  for (j=1; j<=m; j++)
    if (X[i][j] == T) V= V+1;

```

- А) сумма элементов главной диагонали матрицы;
- Б) сумма элементов каждой строки матриц X и Y;
- В) количество элементов матрицы равных T;
- Г) количество равных элементов матрицы X;

8. Что определяет для массива $X[m][m]$ следующий алгоритм

```

for (i=1; i<=m ; i++)

```

```

for (j=1; j<=n ;j++)
    if (i == j) X[i][j] =1;

```

- А) формирование матрицы, в которой элементы стоящие в нечетных столбцах равны 1, остальные равны 0;
- Б) формирование матрицы, в которой элементы стоящие на главной диагонали равны 1, остальные равны 0;
- В) формирование матрицы, в которой элементы стоящие в четных строках равны 1, остальные равны 0;
- Г) количество равных элементов матрицы X;

9. Что определяет для массива X[n][m] следующий алгоритм

```

for (i=1; i<=n ; i++)
    for (j=1; j<=m; j++)
        if (i % 2 == 0) X[i][j] =1;

```

- А) формирование матрицы, в которой элементы стоящие в нечетных столбцах равны 1, остальные равны 0;
- Б) формирование матрицы, в которой элементы стоящие на главной диагонали равны 1, остальные равны 0;
- В) формирование матрицы, в которой элементы стоящие в четных строках равны 1, остальные равны 0;
- Г) количество равных элементов матрицы X;

10. Что определяет для массива X[n][m] следующий алгоритм

```

for (i=1; i<=n ; i++)
    for (j=1; j<=m; j++)
        if (j % 2 != 0) X[i][j] =1;

```

- А) формирование матрицы, в которой элементы стоящие в нечетных столбцах равны 1, остальные равны 0;
- Б) формирование матрицы, в которой элементы стоящие на главной диагонали равны 1, остальные равны 0;
- В) формирование матрицы, в которой элементы стоящие в четных строках равны 1, остальные равны 0;
- Г) количество равных элементов матрицы X;

Подпрограммы пользователя. Задачи.

1. Описать подпрограмму, которая вычисляет факториал. Вычислить значение следующего выражения:

$$\frac{a! + (a+b)!}{c! - (a+c)!} - (2a+1)!$$

2. Описать подпрограммы, которая вычисляет максимальное и минимальное значение двух чисел. Вычислить значение следующего выражения:

$$\frac{\max^2(a,b,c) - \min(a,b,c)}{\max(a,b) - \min(b,c)}$$

3. Описать подпрограмму нахождения НОД(a,b) двух чисел и вычислить НОД(a,b,c).
4. Ввести с клавиатуры 3 любых целых числа. Найти среднее арифметическое максимальных цифр этих чисел.
5. Создать 3 массива А, В, С: размерности 10, 15 и 20, диапазоны значений элементов массивов: [-10..10], [-15..15], [-20..20]. Найти сумму минимальных элементов массивов. Создание массива оформить в виде подпрограммы-процедуры, поиск минимального элемента в массиве – в виде подпрограммы-функции.

6. Преобразовать строку так, чтобы буквы каждого слова в ней были отсортированы по алфавиту.
7. Написать программу, которая переводит число из десятичной системы счисления в двоичную или восьмеричную.
8. Число, представленное в шестнадцатеричной системе счисления, перевести в десятичную систему счисления.

5

Оператор множественного выбора



Существует множество задач, при решении которых возникает необходимость из множества возможных вариантов выбрать один. В этом случае используется оператор выбора, который является обобщением условного оператора и предназначен для замены конструкции из вложенных операторов if...then.



форма записи оператора множественного выбора



```

switch (/*переменная или выражение*/)
{
  case const1:
  {
    /*группа операторов*/;
    break;
  }
  case const2:
  {
    /*группа операторов*/;
    break;
  }
  .....
  default:
  {
    /*группа операторов*/;
  }
}
    
```





Порядок работы оператора выбора:



Значение выражения или переменной, стоящей после служебного слова **switch**, последовательно сравнивается со значениями констант `const1`, `const2` и т.д. Если найдено совпадение значений, то выполняется оператор или группа операторов, помеченная этой константой, пока не встретится зарезервированное слово **break** или закрывающая фигурная скобка. Если значение переменной или выражения не совпадает ни с одной из констант, то передается управление ветви программы содержащей зарезервированное слово **default (иначе)**. После чего выполняется оператор или группа операторов данной ветви.



Примечание

- ❖ переменная либо выражение стоящие после `switch` имеют только порядковый (целый либо символьный) тип;
- ❖ тип констант должен совпадать с типом выражения;
- ❖ ветвь **default** является необязательной;

Пример:

```
switch (S)
{
    case 1: {
        a++;
        b=0;
        break;
    }
    case 5 case 6: {x=-1; break;}
    default: x=2;
}
```

- ❖ Все константы в списках должны быть уникальны в пределах оператора выбора, диапазоны не должны пересекаться и не должны содержать констант, указанных в данном или других списках.
- ❖ Если значение переменной или выражения по условию задачи может совпадать со значениями нескольких констант, то их можно перечислить через пробел например, `case1: case2: case3: {a++; break;}`
- ❖ Операторные скобки `{ }` в `case` могут отсутствовать



Условие задачи: написать программу, которая складывает, вычитает, умножает, делит два числа введенных с клавиатуры. Разработать пользовательский интерфейс.



```
#include <iostream>
using namespace std;
int main()
{
    int count;           // переменная для выбора в switch
    double a,b;         // переменные для хранения операндов
    cout << "Vvedite pervoe chislo: ";
    cin >> a;
    cout << "Vvedite vtoroe chislo: ";
    cin >> b;
    cout << "Vibirite deistvie: 1-clojenie; 2-vichitanie; 3-ymnojenie; 4-delenie: ";
    cin >> count;
    switch (count)      // начало оператора switch
    {
        case 1:        // если count = 1
        {
            cout << a << " + " << b << " = " << a + b << endl; // выполнить сложение
            break;
        }
        case 2:        // если count = 2
```

5



```
{
    cout << a << " - " << b << " = " << a - b << endl; // выполнить вычитание
    break;
}
case 3: // если count = 3
{
    cout << a << " * " << b << " = " << a * b << endl; // выполнить умножение
    break;
}
case 4: // если count = 4
{
    cout << a << " / " << b << " = " << a / b << endl; // выполнить деление
    break;
}
default:
    // если count равно любому другому
    значению
    cout << "Nepравilni vvod" << endl;
}
return 0;
}
```



Программирование на языке C++

Операторы цикла (заключительное занятие)



Циклы

Цикл – это многократное выполнение одинаковой последовательности действий.

- цикл с известным числом шагов
- цикл с неизвестным числом шагов (цикл с условием)



2

Операторы цикла.

Вопросы:

1. В каких случаях используются операторы цикла?
2. Может ли тело оператора цикла с параметром не выполниться ни разу?
3. Почему в программировании существует правило: нельзя изменять параметр цикла в теле цикла?
4. Как записывается команда цикла, в которой счетчик цикла уменьшается?
5. Можно ли изменять значение верхней границы цикла в теле цикла?
6. Что такое «структура вложенных циклов»?
7. При вложении какой цикл будет внутренним?
8. Как работает структура вложенных циклов.



3

Сколько раз выполняется цикл?

```
a = 1;  
for (i=1; i<=3; i++)  
  a = a+1;
```

```
a = 1;  
for(i=6; i<=2;i++) a = a+1;
```

```
a = 1;  
for(i=1 ;i>=3; i--) a = a+1;
```

```
a = 1;  
for (i=3; i>=1;i--) a = a+1;
```

```
a = 1;  
for (i=3 ;i<= 3;i++) a=a+1;
```



4

Оператор цикла с предусловием.

Какой результат мы получим в представленных фрагментах программ?

```
1) x=1;
   while (x>1)
       x=x+1;
```

```
2) x=10;
   while (x>1)
       x=x+1;
```

```
3) x=2; y=3;
   while (x<=5)
       x=x+1;
       y=y+2;
   cout<<x," ",y;
```

```
4) x=1; y=1;
   while (x<10)
   {
       y=y+1;
       x=x+1;
   }
```



5

Оператор цикла с постусловием.

Какое значение примет переменная x в результате выполнения следующих фрагментов программ?

```
1) x=1;
   do
       x=x+1;
   while (x<=10);
```

```
2) x=1;
   do
   { x=x+3;
     x=x+1;}
   while (x>10);
```

```
3) x=1; y=1;
   do
   {y=y+1;
     x=x+1;}
   while (x=10);
```

```
4) x=10; y=10;
   do
   {y=y+1;
     x=x+1;}
   while (x<10);
```



6

Оператор цикла с параметром.

В результате выполнения программы значение переменной x должно стать равным **100**. Выберите правильный вариант из приведённых ниже программ. В остальных программах укажите строки, в которых, по вашему мнению, содержится ошибка.

```
1) int i, x;
x=0;
for (i=1; i<=100; i++)
    x=x+1;
```

```
2) int i, x;
x=0;
for (i=1; i<=100; i++)
    x=x+i;
```

```
3) x=1; y=1;
do
    {y=y+1;
    x=x+1;}
while (x=10);
```

```
4) real i, x;
x=0;
for (i=1; i<=100; i++)
    x=x+1;
```

7

Структура вложенных циклов

Цикл называется **вложенным**, если он находится в теле другого цикла.

```
for (i=0; i<n; i++)
{
    for (j=0; j<m; j++)
    {
        тело цикла
    }
}
```

Внутренний цикл должен полностью находиться в теле внешнего цикла.

8

Циклы_С++ [Режим совместимости] - PowerPoint

Файл Главная Вставка Дизайн Переходы Анимация Слайд-шоу Рецензирование Вид Что вы хотите сделать?

С С текущего слайда Очередная презентация Произвольное слайд-шоу Начать слайд-шоу

Настройка слайд-шоу Скрыть слайд-шоу слайд

Настройка времени слайд-шоу

Настройка

Воспроизвести речевое сопровождение

Использовать время

Показать элементы управления проигрывателем

Мониторы Автоматически

Режим докладчика

Мониторы

Задачи

- Ввести любое натурально число. Найти все его делители.
- Среди всех трехзначных чисел найти такие, в которых зачеркнули первую цифру слева, полученное двузначное число умножили на 9, то получили исходное число.
- Среди всех двузначных чисел найти те, сумма квадратов цифр которых делится на 13.
- Найти цифровой корень введенного с клавиатуры натурального числа (цифровой корень – сумма цифр числа меньше 10).
- Даны натуральные числа n и m . Найти все пары дружественных чисел, лежащих в диапазоне от n до m . Два числа называются дружественными, если каждое из них равно сумме всех делителей другого (само число в качестве делителя не рассматривается).

Заметки к слайду

Слайд 9 из 9 русский

Секретно Привлечение

12:52 06.05.2019