

Министерство образования и науки Хабаровского края

Краевое государственное бюджетное образовательное учреждение дополнительного образования
«Хабаровский краевой центр развития творчества детей и юношества»

Центр технического творчества

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

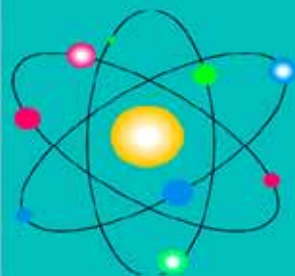
Математика



Информатика



Физика



в школах Хабаровского края

МИФ

№ 1 (5) 2018 г.

Хабаровск
2018

Печатается по решению
научно-методического совета
КГБОУ ДО ХКЦРТДиЮ
протокол № 4 от 04.12.2017 г.

МИФ: математика, информатика, физика в школах Хабаровского края. Методические рекомендации / Сост. Н.М. Борисова. – Хабаровск: КГБОУ ДО ХКЦРТДиЮ, 2017. – 44 с.

В данных методических рекомендациях представлены статьи педагогов высшего профессионального образования и методиста центра технического творчества Хабаровского краевого центра развития творчества детей и юношества по математике, физике и информатике.

Методические рекомендации адресованы педагогическим работникам, осуществляющим физико-математическое обучение детей.

Ответственный редактор: В.В. Лежнина

Ответственный за выпуск: А.Ф. Немцев, директор ЦТТ КГБОУ ДО ХКЦРТДиЮ

Научные консультанты:

по математике: О.В. Тулицына, учитель математики,

С.В. Горностаева, учитель информатики и ИКТ МБОУ СОШ № 6 г. Бикина;

по физике: Л.В. Горбанева, старший преподаватель кафедры физики

ПИ ФГБОУ ВО ТОГУ, г. Хабаровск;

по информатике: Е.А. Редько, старший преподаватель кафедры математики

и информационных технологий ФГБОУ ВО ПИ ТОГУ, г. Хабаровск

Компьютерная верстка: К.И. Спика

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	2
Л.В. Маруга. На пути вступления в ФГОС старшей школы.....	3
МАТЕМАТИКА	
О.В. Тупицына, С.В. Горностаева. Практико-ориентированный проект: «Разработка тренажёра для решения уравнений в системе объектно- ориентированного программирования «Embarcadero RAD Studio XE3.....	7
ИНФОРМАТИКА	
Е.А. Редько. Методические рекомендации по освоению базовых навыков программирования на языке С. Часть II	19
ФИЗИКА	
Л.В. Горбанева. Самодельный прибор для определения изменения погоды.....	37
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	41

ВВЕДЕНИЕ

В методических рекомендациях представлены статьи педагогов высшего профессионального образования, а так же учителей общеобразовательных школ по математике, физике и информатике. В них рассматриваются различные аспекты проектно-исследовательской деятельности обучающихся, представлен материал по программированию и освоению средств языка Си, реализующих базовые алгоритмические конструкции, практико-ориентированный проект: «Разработка тренажёра для решения уравнений в системе объектно-ориентированного программирования «Embarcadero RAD Studio XE3», предложен алгоритм изготовления приборов, позволяющих обучающимся определить погодные условия.

Материалы, представленные в методических рекомендациях, будут интересны педагогам при организации различных занятий, направленных на применение знаний обучающимися, при решении задач в контексте различных жизненно-практических ситуаций.

На пути вступления в ФГОС старшей школы

Среднее общее образование – третий, завершающий уровень общего образования. Данный уровень общеобразовательной школы в процессе модернизации системы образования подвергается самым существенным структурным, организационным и содержательным изменениям. Социально-педагогическая суть этих изменений – обеспечение наибольшей личностной направленности и вариативности образования, его дифференциации и индивидуализации. Эти изменения являются ответом на социальный заказ – максимально раскрыть индивидуальные способности, дарования человека и сформировать на их основе профессионально и социально компетентную, мобильную личность, умеющую делать профессиональный и социальный выбор и нести за него ответственность, сознающую и способную отстаивать свою гражданскую позицию, гражданские права.

Выстраивание образовательного пространства в МБОУ СОШ № 6 г. Бикина, цифровой школе старшей ступени обучения, где учатся школьники только 10–11 классов, создаются условия для социального и образовательного самоопределения старшеклассников, для получения школьниками качественного современного образования, позволяющего выпускнику занимать осмысленную, активную и деятельную жизненную позицию, поступить и успешно обучаться в выбранном вузе.

Для этого в школе создано многопрофильное обучение. Ведущими профилями в школе стали физико-математическое, химико-биологическое и социальное направления. Они определяют формирование профильных классов. Летом организуются социальные практики и профессиональные пробы для учащихся на предприятиях и в различных организациях города, где каждый десятиклассник может пройти практику и в дальнейшем сделать правильный выбор профессии.

В старшем подростковом возрасте (15–17 лет) ведущую роль играет учебно-профессиональная деятельность по овладению

системой научных понятий в контексте предварительного профессионального самоопределения. У старших подростков сохраняет своё значение учебная деятельность, стремление к саморазвитию и самообразованию, поэтому в 2016–2017 учебном году для учащихся десятых классов в качестве переводного экзамена была определена форма защиты проекта или исследовательской работы. Мы считаем, что эксперимент прошел хорошо. Однако эта работа не была бы успешной, если бы мы не ввели ещё одно новшество, о котором хотели бы рассказать.

На протяжении трех лет в сентябре в школе проводится Неделя погружения в проектно-исследовательскую деятельность. Дело в том, что школьники в МБОУ СОШ № 6 г. Бикина, цифровой школе старшей ступени, обучаются всего 2 года. Придя в эту школу, они должны быть готовы к организации самостоятельной деятельности, в том числе и проектной, и исследовательской. Внедрение практики по реализации ФГОС III поколения осуществляется через активное и стремительное включение школьника, старшеклассника в максимально самостоятельную образовательную деятельность через проектно-исследовательскую деятельность, которая осуществляется в рамках одного или нескольких учебных предметов с целью демонстрации своих достижений.

Именно поэтому в первую неделю сентября учебного года организуется Неделя погружения в проектно-исследовательскую деятельность. В проекте принимают участие все учащиеся школы.

Основной целью деятельности всех участников образовательного процесса (учащихся, педагогов, родителей) при реализации данного проекта, основанного на принципах внедрения личностно-ориентированных и развивающих технологий, является формирование интереса у ребят к проектно-исследовательской и творческой деятельности, создание собственного продукта и умение его защищать.

Учебно-исследовательская и проектная деятельность организована в школе таким образом, чтобы учащиеся могли формировать ключевые компетенции через сотрудничество в творческих группах одноклассников, учителей и т.д. Строя различного рода отношения в ходе целенаправленной поисковой, творческой и продуктивной деятельности, подростки овладевают нормами взаимоотношений с разными людьми, учатся переходить от одного

вида общения к другому, приобретают навыки индивидуальной и коллективной деятельности.

Одной из особенностей Недели погружения является самооценивание хода и результата работы. Это позволяет, оглянувшись назад, увидеть допущенные просчёты (на первых порах это переоценка собственных сил, неправильное распределение времени, неумение работать с информацией, вовремя обратиться за помощью), проанализировать недостатки и скорректировать деятельность.

Педагоги школы при сопровождении данного проекта выступают в роли тьюторов, учитывая следующие моменты:

- тема исследования должна быть на самом деле интересна для ученика и совпадать с учетом компетентности учителя, эти темы порой становятся темами социальных проектов, имеющих практическую направленность;

- необходимо, чтобы учащийся четко осознавал суть проблемы, иначе весь ход поиска её решения будет бессмысленен, даже если он будет проведён учителем безукоризненно правильно;

- организация хода работы над раскрытием проблемы исследования строится на взаимответственности учителя и ученика друг перед другом и взаимопомощи;

- раскрытие проблемы, в первую очередь, должно приносить что-то новое ученику.

В рамках проведения этой недели мы создаем для учащихся условия, позволяющие рассматривать социальные, экономические, познавательные проблемы, а метод проектов предполагает организацию таких ситуаций, в которых учащиеся решают значимые для себя проблемы, что позволяет десятиклассникам успешнее адаптироваться и включиться в процесс обучения в школе старшей ступени.

Дальнейшая проектно-исследовательская деятельность учащихся продолжается в течение всего учебного года на базе школьного научного общества «Алые паруса». Результаты работы представляются на школьной научно-практической конференции «Факел знаний и открытий», что является вторым этапом реализации проекта.



Подобная организация образовательной деятельности, по мнению педагогического коллектива, отвечает требованиям ФГОС старшей школы. В число таких требований входят компетентности, связанные с идеей опережающего развития (все то, что понадобится школьникам и в дальнейшем образовании, и в будущей взрослой жизни, чтобы ориентироваться в постоянно меняющемся мире, успевать осваивать расширяющееся

информационное пространство). Результатом деятельности школы старшей ступени должен стать выпускник, способный конкурировать в этом мире.

Особенно интересными были проекты по физике, математике, астрономии, физике, а также метапредметные проекты – на стыке наук физики и биологии, информатики и математики, астрономии и математики. Под руководством учителей-кураторов Дубинской О.В., Веренич Н.П., Кожухаренко В.И., Маруга Л.В., Плоховой О.М., Шевелевой Е.П., Шишкиной К.В., Кужим С.В., Горнастаевой С.В., Тупицыной О.В., Балыковой Е.М., Яновой Н.А. ребята выполнили и успешно защитили свои исследовательские работы и проекты: «Макет Солнечной системы», «НЛО – реальность или вымысел?», «Мифы зодиакальных созвездий», «Плутон и Харон», «Черные дыры» и многие другие.

Результаты проведенной работы показали наличие желания и потребности у обучающихся в углубленном изучении избранной области знаний, их самостоятельном добывании и готовности к постоянному самосовершенствованию. И это еще одна ступень на пути вступления в ФГОС среднего общего образования, время которого неумолимо приближается.



МАТЕМАТИКА

О.В. Тупицына, учитель математики,
С.В. Горностаева, учитель информатики и ИКТ
МБОУ СОШ № 6 г. Бикина

Практико-ориентированный проект: «Разработка тренажёра для решения уравнений в системе объектно-ориентированного программирования «Embarcadero RAD Studio XE3»

Авторы проекта:

Дорошенко Кирилл, учащийся 11 класса,
Ефимович Савелий, учащийся 10 класса.

1. Введение

*«Ум человеческий только тогда понимает обобщение,
когда он сам его сделал или проверил»
Л.Н. Толстой*

1.1. Актуальность выбранной темы проекта.

Вопрос о способах решения продвинутых математических уравнений, встречающихся при изучении математики, которые сводятся к квадратным путём замены неизвестного всегда вызывает особые затруднения.

В практике встречается множество различных видов уравнений, к каждому из которых требуется применить исключительно свою собственную подстановку. Трудности образуются буквально с первых шагов исследования данного уравнения. Достаточно трудоёмко и затратно по времени определяются пути решения. Если владеть знаниями о видах такого типа уравнений и иметь навык видеть, какую замену неизвестного следует выполнить, проверить решение с помощью компьютерной программы, то действия становятся проще, очевидней и приводят к успешно выполненной задаче.

1.2. Итак, **проблема** очевидна. Сформулируем возникшие **проблемные вопросы**: Как в каждом из множества различных уравнений увидеть очевидные и неочевидные замены неизвестного? Какое модульное программирование использовать для создания тренажёра?

Поэтому мы решили создать практико-ориентированный проект, состоящий из трёх основных частей: распределили

и проклассифицировали множество уравнений по видам замены целого выражения на подстановку другой неизвестной; составили пошаговые модули для решения пока трёх видов уравнений.

1.3. Идея практико-ориентированного проекта: Как, используя учебную и справочную литературу, интернет-ресурсы, возможности доступного программного обеспечения, создать учебный тренажёр – комплекс, с помощью которого можно решить уравнение, проверить верность выполняемых шагов и сверить полученные результаты корней уравнения со своим решением.

1.4. Цель проекта:

Создание тренажера, позволяющего на основе классификации уравнений, сводящихся к квадратным путём замены неизвестного по виду подстановки, в среде программирования **EmbarcaderoRADStudioXE3**.

1.5. Задачи проекта:

1. Изучить специальные источники информации по заданной теме.

2. Провести мониторинговую работу для исследования у сверстников, имеющих интерес к изучению математики по вопросам изучения данной темы.

3. Проклассифицировать уравнения, сводящиеся к квадратным путём замены неизвестного по виду замены.

4. Изучить структуру и возможности программ **EmbarcaderoRADStudio**.

5. Построить план шагов для построения тренажёра.

6. Построить модель тренажёра для решения уравнений.

7. Опробовать на практике готовый тренажёр.

8. Подготовить 1 часть методической разработки «Тренажёр для решения уравнений, сводящихся к квадратным заменой неизвестной», учитывая рекомендации по улучшению качества продукта.

9. Максимально раскрыть собственный творческий потенциал при создании проекта.

1.6. Время реализации:

Наш проект долгосрочный, рассчитан на два года. Разрабатываем его мы, ученики 10 и 11 классов.

1.7. Ожидаемый результат:

Для себя в самореализации через создание проекта должны получить: мотивацию исследовательской деятельности; творческую активность; самостоятельный поиск; познавательный интерес к изучению разделов математики и компьютерных учебных программ; исследовательские, информационные, коммуникативные умения;

реальный пока незаконченный результат в виде тренажёра; более глубокие познания в области решения уравнений, качественная подготовка к ЕГЭ.

Обобщение материала по решению уравнений служит: усовершенствованию в математическом познании в узком направлении и освоению математики в целом; качественной подготовке к сдаче экзаменов.

Для кабинетов математики и информатики: готовый образовательный продукт в виде учебного пособия.

1.8. Этапы работы над проектом:

1. Изучение учебной, справочной литературы, интернет-ресурсов и готового банка задач, подобранных одноклассниками на уроках алгебры по теме «Решение уравнений, сводящихся к квадратным путём замены неизвестного».

2. Сбор, обработка и анализ материала, касающегося интереса, трудностей и полезности изучения темы.

3. Классификация уравнений по виду замены неизвестной.

4. Изучение структуры и возможностей используемой программы.

5. Построение плана шагов и модели тренажёра.

6. Наглядное представление готового продукта деятельности.

7. Выводы по проведенному исследованию и перспективы проекта.

1.9. Основное содержание проекта.

Данный проект состоит из трёх разделов. Каждая часть представляет собой законченный блок (обобщение теоретического материала, разработка алгоритма и написание программы «тренажер», тестирование работы программы). Создание тренажёра – это центральный блок в проекте.

2. Классификация уравнений, сводящихся к квадратным путём замены неизвестного по виду замены

2.1.1. Еще в древности одним из важнейших достоинств человека считали владение математическими знаниями.

Недостаточно лишь понять задачу, необходимо желание решить ее.

Без сильного желания решить трудную задачу невозможно, но при наличии такого возможно. Где есть желание, найдется путь! (Пойа Д.).

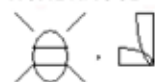
Термин «алгебра» как один из разделов математики в целом, в древности же у арабов имел название «искусства восстановления», то есть переходил в медицину. Вправленные кости ломаной руки или ноги также являлось восстановлением потерянного органа, и искусство

врача, которое возвращает человеку руку или ногу, также называлось алгеброй. Такой двойной смысл слова «алгебра» объясняет нам один странный, на первый взгляд, факт. Все вы знаете известный роман Сервантеса «Дон Кихот», в котором в 15 главе рассказывается, как Дон Кихот сбил с лошади своего противника, как тот лежал на земле, не будучи в состоянии шевелить ни руками, ни ногами, и как Дон Кихоту удалось найти алгебраиста для оказания помощи побежденному противнику. Так сказано в испанском оригинале романа, так же говорится в более ранних русских изданиях этого романа; только в последнем издании «алгебраист» заменен на «костоправ». Объясняется это тем, что в испанском и португальском языках слово «алгебра», как и в арабском языке, означает не только часть математики, но и искусство вправлять вывихи; словом «алгебраист» называется не только знающий алгебру, но и врач – специалист по болезням рук и ног.

Точно так же, как алгебраист возвращал больному потерянный орган, уравнение требует нахождения своих корней.

Так что это за «диво» – уравнение?

Уравнение от любого другого выражения отличается тем, что в нем есть буквы, знак равенства. Употребление букв в алгебре появилось в результате очень долгого развития. Особый знак



, (назывался он хау, что в переводе на русский язык «куча») был у египтян. Индусские математики при решении уравнений, получив отрицательный результат, толковали его как долг или расход и обозначали точкой над числом или крестиком рядом с ним. Отрицательные числа с трудом проникают в математику. К ним математики подошли при решении уравнений, когда возникали случаи вычитания из меньшего числа большего.

Окончательно вводит в математику отрицательные числа Рене Декарт, который дает геометрическое истолкование и определяет место и порядок следования на числовой оси.

О символике: математики, писавшие на арабском языке, в том числе и среднеазиатские, неизвестное искомое число называли «вещью». Первая буква этого слова в европейской транскрипции и дала нам обозначение неизвестного буквой x .

Искусство решать уравнения зародилось у вавилонян, у которых для него было специальное название, перешедшее в арабский язык. В рассказе о вавилонской математике было уже сказано, что

вавилоняне решали уже уравнения 1-ой и 2-ой степени, а при помощи таблиц – и некоторые виды уравнений 3-ей степени. Узбекский математик Аль-Хорезми свою книгу начала IX века, которая, переведенная в XVII веке на латинский язык, стала родоначальником европейских учебников алгебры, называет «Китаб-ал-джабр вал-мукабала», что в переводе означает «Книга о восстановлении и противопоставлении». «Восстановление» означает превращение вычитаемого (по современному «отрицательного») числа в положительное при перенесении из одной половины уравнения в другую. Так как в те времена отрицательные числа не считались настоящими числами, то операция «ал – джабр» (алгебра), как бы возвращающая число из небытия в бытие, казалась чудом этой науки, которую в Европе после этого называли «великим искусством» рядом с «малым искусством» – арифметикой.

2.1.2. Итоги тестирования учащихся физико – математического профиля

Всего было опрошено 25 учащихся 10–11 классов физико-математического профиля.

Для опроса нами были составлены следующие вопросы:

1. Все ли темы алгебры для собственного изучения тебе интересны?
2. Любишь ли решать сложные уравнения?
3. Что вызывает затруднения при решении уравнений?
 - а) Незнание свойств функций, входящих в уравнения.
 - б) Незнание видов подстановки, заменяющей неизвестную.
 - в) Неумение заменять целое выражение одной переменной.
4. Хотел ли ты, чтобы использовались специальные тренажёры, позволяющие проверять решения уравнений?

При опросе были получены следующие результаты:

- 1) По 1, 2 и 4 вопросу результаты представлены на 1 диаграмме.



2) По 3 вопросу результаты представлены на 2 диаграмме.



Виды уравнений:

В 10 классе в курсе алгебры и начал математического анализа на учебных занятиях в мини-проектах, работая коллективно в группах, нами был подобран банк задач по каждому типу уравнений отдельно каждой из шести групп. Работая же над самим проектом, мы собрали и обобщили весь собранный материал, вывели общие формулы по каждому типу уравнений и с помощью формул составили запись целых выражений, которые в общем виде имеют возможную замену неизвестной, то есть проклассифицировали их в общем виде. Что имеет принципиальное отличие от сформированного банка задач, состоящих из примеров конкретных уравнений.

2.2.1. *Дробно-рациональные уравнения общего вида с подобранными примерами.*

Уравнение вида $\frac{d}{ax^2+bx+c} + \frac{d}{ax^2+bx+k} = L,$

Область определения уравнения: $ax^2 + bx + c \neq 0$ и $ax^2 + bx + k \neq 0$

Пусть $t = ax^2 + bx$, но при сведении его к квадратному, необходимо указать, каким не может быть t .

Примеры данного типа уравнений:

$$1) \frac{3}{x^2-4x+1} - x^2 = 3-4x; \quad 2) \frac{16}{(x+6)(x-1)} - \frac{20}{(x+2)(x+3)} = 1;$$

$$3) \frac{4x^2+7x+7}{8x^2+14x+10} + \frac{4x^2+7x+1}{12x^2+21x+13} = \frac{1}{2}; \quad 4) 9x - 6x^2 = \frac{2x^2-3x-4}{2x^2-3x+1} = 4.$$

2.2.2. Алгебраические уравнения с очевидной заменой неизвестного.

Уравнение вида: $(ax^2 + bx + c)(ax^2 + bx + d) = k$, где $t = ax^2 + bx$,

О.О.У: $x \in R$, но и $t \in R$ и сведение его к квадратному или вида

$(x+a)(x+b)(x+c)(x+d) = L$, такая же замена $t = ax^2 + bx$.

Примеры данного типа уравнений:

$$1) (x+1)^2(x^2+2x) = 12; \quad 2) (x^2+3x+1)(x^2+3x+3) + 1 = 0;$$

$$3) (x^2-2x)^2 - 3x^2 + 6x - 4 = 0; \quad 4) (x^2 - 5x + 7)^2 - (x-3)(x-2) - 1 = 0;$$

$$5) (x-4)(x-3)(x-2)(x-1) = 24; \quad 6) x(x+4)(x+5)(x+9) + 96 = 0.$$

2.2.3. Симметрические (возвратные и невозвратные) уравнения.

Уравнения вида: $ax^n + a_1x^{n-1} + \dots + a_1x + a_0 = 0$.

Рассматриваются случаи:

а) когда n – чётное число, тогда уравнение является *возвратным*, подстановка здесь $t = x + \frac{1}{x}$ и сводится данное уравнение к квадратному путём деления каждого члена на центральный член уравнения.

б) когда n – нечётное, тогда один корень всегда известен сразу, то есть $x_1 = -1$, затем, понижая степень уравнения путём деления на выражение $(x+1)$, сведение такого уравнения также приводит к возвратному и замена будет такая же, как и у возвратного, то есть $t = x + \frac{1}{x}$.

Примеры данного типа уравнений:

1) $x^4 - 7x^3 + 14x^2 - 7x + 1 = 0$; 2) $2x^4 + x^3 - 11x^2 + x + 2 = 0$;

3) $2x^3 - 7x^2 - 7x + 2 = 0$; 4) $-2x^5 - 4x^4 - 2x^3 - 2x^2 - 4x - 2 = 0$.

2.2.4. Однородные уравнения.

Уравнения вида: $y^{2a}(x) + z^a(x)y^a(x) + z^{2a}(x) = 0$.

Чтобы привести уравнение данного вида к квадратному уравнению с очевидной заменой неизвестного, необходимо разделить все члены уравнения на одну из функций с наивысшей степенью, входящей в данное уравнение, например на $z^{2a}(x) > 0$, так как имеет место показательная функция и получается уравнение вида: $(\frac{y}{z})^{2a}(x) + (\frac{y}{z})^a(x) + 1 = 0$.

Примеры данного типа уравнений:

1) $(x+5)^4 - 13x^2(x+5)^2 + 36x^4 = 0$; 2) $2(x-1)^4 - 5(x^2 - 3x + 2)^2 + 2(x-2)^4 = 0$;

3) $(x^2 - 3x + 1)^2 + 3(x-1)(x^2 - 3x + 1) = 4(x-1)^2$; 4) $2(x^2 + x + 1)^2 - 7(x-1)^2 = 13(x^3 - 1)$.

2.2.5. Уравнения. Решаемые методом выделения полного квадрата.

Уравнения вида: $x^2 + \frac{x^2}{(x+a)^2} = b$, здесь левая часть представляет собой часть формулы квадрата суммы или разности чисел, поэтому при решении таких уравнений, можно использовать метод выделения полного квадрата, с помощью формулы $(a \pm b)^2 \mp 2ab = a^2 + b^2$. Тем самым также приходим к очевидной замене неизвестной.

Примеры данного типа уравнений:

1) $x^2 + \frac{x^2}{(x+1)^2} = 3$; 2) $x^2 + \frac{4x^2}{(x+2)^2} = 5$; 3) $x^2 + \frac{9x^2}{(x-3)^2} = 7$;

4) $x^2 + \frac{25x^2}{(5+2x)^2} = \frac{74}{49}$.

2.2.6. Уравнения, сводящиеся к биквадратным.

Уравнения вида: $(x+a)^4 + (x+b)^4 = c$, где $x = t - \frac{a+b}{2}$, и приведение уравнения к более простому виду $(t-1)^4 + (t+1)^4 = c$. Затем, раскрывая четвёртую степень разности и суммы чисел t и 1 , путём $(t-1)^2(t-1)^2$

и $(t+1)^2(t+1)^2$ (можно воспользоваться формулой Герона) приходим к биквадратному уравнению с переменной t .

Примеры данного типа уравнений:

$$1) (x+3)^4 + (x+5)^4 = 16; 2) (x+\frac{3}{2})^4 + (x-\frac{1}{2})^4 = 82; 3) (x+2)^4 + x^4 = 82.$$

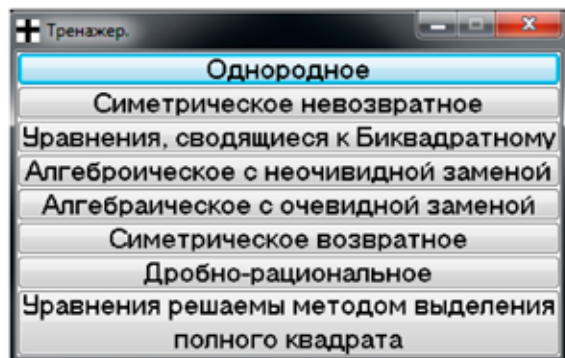
В ходе данной работы мы применили поисково-исследовательскую деятельность. Классификация уравнений и данный банк задач может быть использован как на уроках математики для освоения данного материала, так и для подготовки к ЕГЭ. Нами был накоплен собственный опыт поиска, структурирования и исследования в ходе выбора правильного пути для решения того или иного уравнения.

Создание тренажёра для решения уравнений в программе EmbarcaderoRADStudioXE3.

Без этой части работы невозможно создание тренажёра, позволяющего решать такие типы уравнений, однако проект, который мы разрабатываем, заключается именно в использовании данного материала. Акцент сделан на использовании такой компьютерной программы, которая позволяет разработать тренажёр для решения уравнений в системе объектно-ориентированного программирования «Embarcadero RAD Studio XE3. Это и является основным в нашей работе.

2. Этапы разработки программы.

Всем хорошо известно такое расхожее выражение, как «написать программу». Однако, написание программы – это всего лишь один из этапов создания программы. Программирование – это процесс разработки или создания программы,



который очень удобно рассматривать как последовательность следующих шагов:

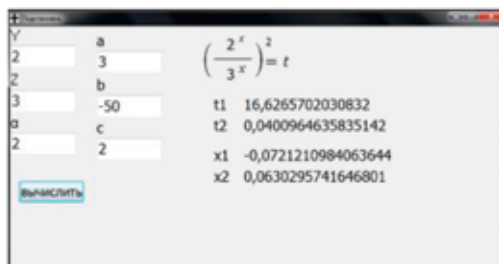
- определение требований к программе,
- разработка или выбор алгоритма решения поставленной задачи,

- написание команд,
- отладка,
- тестирование.

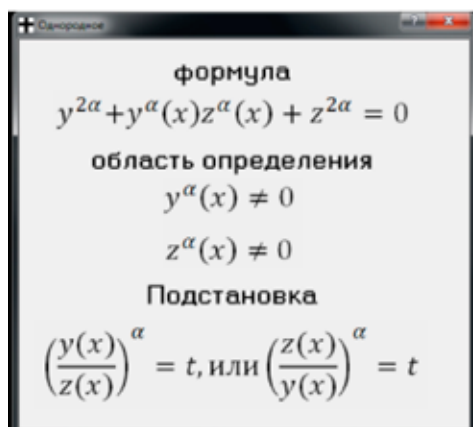
Прокомментируем каждый шаг из указанного перечня действий.

3.1. Определение требований к программе.

Это один из важнейших этапов. На этом этапе подробно описывается исходная информация и формулируются требования к результату. Кроме того описывается поведение программы в особых случаях. Например, при решении квадратного уравнения $ax^2 + bx + c = 0$ требования к программе могут быть такими:



- исходными данными для программы являются коэффициенты a, b, c при степенях неизвестного x, значения которых должны вводиться в режиме диалога во время работы программы;



- выходные данные - значения корней уравнения;
- если уравнение не имеет корней, то должно печататься соответствующее сообщение.

Требования к программе, которая будет работать в системе Windows, могут включать еще пункты о желаемом виде диалоговых окон программы.

3.2. Разработка алгоритма.

На этапе разработки алгоритма необходимо определить последовательность действий, которые надо выполнить для достижения поставленной цели, получения результата. Если задачу можно решать разными способами и, следовательно, возможны различные алгоритмы, то программист, используя некоторый критерий, выбирает наиболее подходящее решение. Затем составляется подробное описание алгоритма.

3.3. Написание команд программы.

Далее как только определены требования к задаче и составлен ее алгоритм, он записывается на выбранном языке программирования.

3.4. Отладка программы.

Под отладкой понимается процесс поиска ошибок в программе и достижения правильной ее работы на специально подобранных контрольных примерах. В программе могут быть синтаксические ошибки (то есть ошибки в тексте) и алгоритмические ошибки. Алгоритмические ошибки обнаружить значительно труднее. Этап отладки считается завершенным, если программа работает на одном-двух наборах входных данных.

3.5. Тестирование программы.

Этап тестирования особенно важен. Ведь Вашей программой вероятно будут пользоваться другие. Поэтому на этом этапе следует проверить работу программы на как можно большем количестве данных. Сюда же нужно включить и заведомо неверные данные. Например, следует проверить, как работает программа вычисления корней квадратного уравнения, если задать равным нулю, например, коэффициент при старшей степени неизвестного x или вообще задать все коэффициенты уравнения нулевыми.

4. Заключение

Наш проект, направлен на решение проблемы обучения, сформулированной нами самими в виде задачи, когда результат этой деятельности – найденный способ решения проблемы – носит практический характер, имеет прикладное значение и, что весьма важно, интересен и значим для нас самих учащихся физико-математического профиля.

Результатом данного проекта будет программное обеспечение, позволяющее увидеть общий вид уравнения и его собственную подстановку, находить корни подстановки и корни самого уравнения в виде методической разработки, который может быть использован в жизни класса, школы, микрорайона, города, государства. Этот ресурс может стать учебным пособием для кабинетов математики и информатики. С помощью него наглядное представление о решениях уравнений, сводящихся к квадратным, могут получить как старшеклассники, так и учащиеся 9-ых классов основной школы. Ещё одним преимуществом данного продукта будет то, что он является

полноценной программой, то есть будет доступен на любом компьютере.

Для себя в самореализации через создание проекта получили:

- мотивацию исследовательской деятельности;
- творческую активность;
- самостоятельный поиск;
- познавательный интерес к изучению учебных программ;
- исследовательские, информационные, коммуникативные умения;
- усовершенствовали свои познания в области решения уравнений;
- систематизировали собственные умения по обобщению учебного материала;
- наработали материал для более качественной подготовки к ЕГЭ и дальнейшего изучения математики;
- реальный результат в виде тренажёра (1 часть).

В заключении хочется процитировать известную притчу:

Шёл мудрец, а навстречу ему три человека, которые везли под горячим солнцем тележку с камнями для строительства. Мудрец остановил их и задал каждому по вопросу. У первого спросил: «Что ты делал целый день?». Тот с ухмылкой ответил, что целый день возил проклятые камни. У второго спросил: «А что ты делал целый день?». Тот ответил: «Я добросовестно выполнял свою работу». А третий улыбнулся, его лицо засветилось радостью и удовольствием. «А я принимал участие в строительстве храма».

Мы сами выбрали эту нелёгкую тему, приложили усилия и потратили немало времени для создания нашего проекта. Хотим сказать, что работа была нам в удовольствие и может быть будет полезна и вам. Как было сказано ранее, работа над второй частью проекта будет продолжена на следующий год.

ИНФОРМАТИКА

*Е.А. Редько, старший преподаватель
кафедры математики и информационных технологий
ФГБОУ ВО ПИ ТОГУ*

Методические рекомендации по освоению базовых навыков программирования на языке Си. Часть II

Методические рекомендации представлены в виде лабораторного практикума, включающего:

- теоретические вопросы, связанные с синтаксисом языка (кратко);
- практические занятия, предусматривающие фронтальное выполнение и разбор готовых примеров;
- задания для индивидуальной работы;
- вопросы для самоконтроля/контроля знаний.

Во второй части методических рекомендаций представлен материал для освоения средств языка Си, реализующих базовые алгоритмические конструкции.

Лабораторная работа № 4. Организация ветвления

Цель: актуализировать понятия логического значения, логической переменной; изучить операции, позволяющие записывать логические выражения; показать форму записи условного оператора, реализующего выбор варианта действий; развивать логическое мышление, алгоритмические способности.

ЛОГИЧЕСКИЙ ТИП ДАННЫХ (BOOL)

Тип `bool` (от слова «boolean» – логический) позволяет вводить в программу переменные, принимающие одно из двух значений: `false` (ложь) или `true` (истина). Данный тип введен в стандарте языка C++, а до этого он определялся на основе целых типов. И по определению значение `false` равно 0 (ноль), а в качестве значения `true` может интерпретироваться любое целое число, отличное от нуля.

Логические переменные широко используются в логических операциях и выражениях.

Размер переменной обычно составляет 2 байта.

Пример объявления:

```
bool flag=false;
```

ОПЕРАЦИИ СРАВНЕНИЯ

== равно или эквивалентно

!= не равно

< Меньше

<= меньше либо равно

> Больше

>= больше либо равно

Пары символов соответствующих операций разделять нельзя.

Общий вид операций отношения (сравнения):

<выражение1> <знак_операции> <выражение2>

Общие правила:

- операндами могут быть любые базовые (скалярные) типы;
- значения операндов после вычисления перед сравнением преобразуются к одному типу;
- результат операции отношения – целое значение 1, если отношение истинно, или 0 в противном случае. Следовательно, операция отношения может использоваться в любых арифметических выражениях.

ЛОГИЧЕСКИЕ ОПЕРАЦИИ

Перечень логических операций в порядке убывания относительного приоритета и их обозначения:

! отрицание (логическое НЕТ)

&& конъюнкция (логическое И)

|| дизъюнкция (логическое ИЛИ)

Общий вид операции отрицания:

!**<выражение>**

Общий вид операций конъюнкции и дизъюнкции:

<выражение1> <знак_операции> <выражение2>

Например:

$y > 0 \ \&\& \ x == 7$ – истина, если 1-е и 2-е выражения истинны, например, при $y=12$ и $x=7$;

$y > 0 \ || \ x == 7$ – истина, если хотя бы одно выражение истинно, например, при $y=-5$, $x=7$.

Ненулевое значение операнда трактуется как «истина», а нулевое – «ложь».

Например:

$!0 \rightarrow 1$

$!5 \rightarrow 0$

Особенность операций конъюнкции и дизъюнкции – экономное последовательное вычисление выражений-операндов. Пусть записана одна из операций: **<выражение1> <операция> <выражение2>**, тогда:

- если выражение1 операции **конъюнкция** ложно, то результат операции – **ноль** и выражение2 не вычисляется;
- если выражение1 операции **дизъюнкция** истинно, то результат операции – **единица** и выражение2 не вычисляется.

Таким образом, появляется возможность – через запись логического выражения задать условную последовательность вычисления выражений в направлении слева направо:

$\text{scanf}(\text{"\%d"}, \&i) \ \&\& \ \text{test1}(i) \ \&\& \ \text{test2}(i) \rightarrow$ нулевой результат одной из функций приведет к игнорированию вызова остальных;

$\text{search1}(x) \ || \ \text{search2}(x) \ || \ \text{search3}(x) \rightarrow$ только ненулевой результат одной из функций приведет к игнорированию вызова остальных.

Пример правильной записи двойного неравенства $0 < x < 100$:
 $(0 < x) \ \&\& \ (x < 100)$.

УСЛОВНЫЕ ОПЕРАТОРЫ

В языке Си имеют отражение обе формы ветвления: полное и неполное. Соответственно получаем два вида условных операторов.

Синтаксис простого (неполного) оператора условного выполнения:

if (выражение) оператор1;

Здесь выражением, как правило, является логическое выражение или выражение отношения. Если выражение в скобках не ноль, т.е. истинно, то выполняется оператор1, иначе он игнорируется. Оператор1 – простой или составной (блок).

Примеры записи:

```
if (x>0) s+=x;
```

```
if (i!=1) j++, s=1; //используем операцию «запятая»;
```

```
    if (i!=1) { j++; s=1; } //последовательность операций в  
                        //операторном блоке;
```

Синтаксис полного оператора условного выполнения:

if (выражение) оператор1; **else** оператор2;

Если выражение в скобках не ноль (истина), то выполняется оператор1, иначе – оператор2. Операторы 1 и 2 могут быть простыми или составными.

Пример записи:

```
if (x>0) j=k+10; else m=i+10;
```

ВЛОЖЕННЫЕ УСЛОВНЫЕ ОПЕРАТОРЫ

Если есть вложенная последовательность операторов if-else, то else связывается с ближайшим предыдущим if, не содержащим else.

Например:

```
if (n>0)
```

```
    if(a>b) z=a; else z=b;
```

Если необходимо связать фразу else с внешним if, то используем операторные скобки:

```
if(n>0)
{if (a>b) z=a;}
else z=b;
```

В следующей цепочке операторов if-else-if выражения просматриваются последовательно:

```
if (выражение1) оператор1;
else
if (выражение2) оператор2;
else
if (выражение3) оператор3;
else оператор4;
```

Если какое-то выражение оказывается истинным, то выполняется относящийся к нему оператор и этим вся цепочка заканчивается. Каждый оператор может быть либо отдельным оператором, либо группой операторов в фигурных скобках. Последняя часть с else имеет дело со случаем, когда ни одно из проверяемых условий не выполняется. Иногда при этом не нужно предпринимать никаких явных действий, в этом случае else <оператор4> может быть опущен, или его можно использовать для контроля, чтобы засечь «невозможное» условие (экономия на проверке условий).

Пример:

```
if (n<0) printf("n отрицательное\n");
else
if (n==0) printf("n равно нулю\n");
else printf("n положительное\n");
```

УСЛОВНАЯ ОПЕРАЦИЯ «? :»

Условная операция – тернарная, в ней участвуют три операнда.

Формат написания условной операции следующий:

```
<выражение1> ? <выражение2> : <выражение3>;
```

Если <выражение1> отлично от нуля (истинно), то результатом операции является <выражение2>, в противном случае – результатом

операции является <выражение3>. Каждый раз вычисляется только одно из выражений 2 или 3.

Запишем оператор if, вычисляющий максимум из a и b и присваивающий его значение z :

```
if (a>b) z=a;  
else z=b;
```

Используя условную операцию, этот пример можно записать:

```
z=(a>b)?a:b;
```

Условную операцию можно использовать также как и любое другое выражение. Если выражения 2 и 3 имеют разные типы, то тип результата определяется по правилам преобразования.

Задача

Дано целое число. Если оно является положительным, то прибавить к нему 1, в противном случае вычесть из него два. Вывести полученное число.

Решение

Так как в задаче требуется выполнить одно из двух возможных действий, будем использовать для ее решения полный условный оператор (if-else). Условие выбора действия зависит от исходного числа, значение которого будем сравнивать с нулем: сравнение $z>0$ принимает истинное значение (true), когда z положительное.

```
#include <stdio.h>  
int main()  
{  
    int z;  
    scanf("%d", &z);  
    if (z>0) z++;  
    else z-=2;  
    printf("z=%d", z);  
    return 0;  
}
```

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Запишите логическое выражение, истинное в том случае, если значение переменной b нечетное.
2. Запишите логическое выражение, истинное в том случае, если значение переменной b четное.
3. Пусть дан фрагмент программы:

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

Определите значение переменной x , которое она получит в результате выполнения этих операторов.
4. Пусть дан фрагмент программы:

```
y = 1; z = 5;  
if (y > 0)  
  if (z > 4) x = 12;
```

Определите значение переменной x , которое она получит в результате выполнения этих операторов. Перепишите вложенный условный оператор, заменив его условным оператором со сложным условием.
5. Пусть дан фрагмент программы:

```
a = 1; b = 0;  
if (a > 0)  
  if (b >= 1) x = 0; else x = 1;
```

Определите значение переменной x , которое она получит в результате выполнения этих операторов. Можно ли переписать вложенный условный оператор, заменив его условным оператором со сложным условием?
6. Пусть дан фрагмент программы:

```
a = 1; b = 0;  
if (a > 0)  
  {if (b >= 1) x = 0; else x = 1;}  
  else x = 2;
```

Определите значение переменной x , которое она получит в результате выполнения этих операторов. Перепишите вложенный условный оператор, заменив его условным оператором со сложным условием.

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

1. Составьте программу, проверяющую, что введенное число является четным или нечетным.
2. Составьте программу, проверяющую, что введенное число делится без остатка на 3.
3. Вы ввели трехзначное число. Затем ввели любую цифру. Составьте программу, определяющую, есть ли в этом числе данная цифра.
4. Составьте программу, определяющую, лежит ли точка с указанными координатами X, Y на окружности радиуса R с центром в начале координат.
5. Составьте программу, определяющую, пройдет ли график функции $y = 5x^2 - 7x + 2$ через заданную точку с координатами (a, b) .
6. В программу передаются результаты соревнований по плаванию для трех спортсменов. Составьте программу, которая выбирает лучший результат и выводит его на экран с сообщением, что это результат победителя заплыва.
7. Прием на работу идет на конкурсной основе. Условия приема требуют 20 лет рабочего стажа и возраста не более 42 лет. Определите, будет ли человек принят.
8. Напишите программу, которая анализирует человека по возрасту и относит к одной из четырех групп: дошкольник, ученик, работник, пенсионер. Возраст человека вводится с клавиатуры.
9. Определите количество знаков в натуральном числе от 0 до 1000.
10. Проверьте, делится ли введенное с клавиатуры натуральное число нацело на 11, 9.
11. Решите уравнение $a|x|=b$.
12. Вводятся три разных числа. Найдите, какое из них является «средним» (больше одного, но меньше другого).
13. Определите четверть координатной плоскости, которой принадлежит точка. Координаты точки ввести с клавиатуры.
14. Определите наибольшее из трех чисел. Числа вводятся с клавиатуры.

Лабораторная работа № 5. Организация циклов

Цель: определить понятие итерации, тела цикла; изучить синтаксис трех видов циклов `for`, `while`, `do..while`; рассмотреть примеры организации циклических конструкций, научиться решать задачи с использованием операторов цикла.

ПОНЯТИЕ ЦИКЛА

Практически все алгоритмы решения задач содержат циклически повторяемые участки. Цикл это одно из фундаментальных понятий программирования. Под циклом понимается организованное повторение некоторой последовательности операторов.

Для **организации** циклов используются специальные операторы.

Любой цикл состоит из:

- **тела** цикла, т.е. тех операторов, которые выполняются несколько раз,
- начальных установок,
- модификации параметра цикла и
- проверки условия продолжения выполнения цикла.

Один проход цикла называется итерацией. Проверка условия выполняется на каждой итерации либо до тела цикла (с предусловием, с параметром), либо после тела цикла (с постусловием).

Перечень разновидностей операторов цикла:

- оператор цикла с предусловием;
- оператор цикла с постусловием;
- оператор цикла с параметром.
-

ОПЕРАТОР ЦИКЛА С ПРЕДУСЛОВИЕМ WHILE

Общий вид:

while (<выражение>) <тело_цикла>;

Если выражение в скобках – истина (не равно 0), то выполняется тело_цикла. Это повторяется до тех пор, пока выражение не примет значение 0 (ложь). В этом случае выполняется оператор, следующий за

while. Если выражение в скобках – ложно (равно 0), то цикл не выполнится ни разу.

Тело_цикла может включать любое количество управляющих операторов, связанных с конструкцией while, взятых в фигурные скобки (блок), если их более одного. Среди этих операторов могут быть continue – переход к следующей итерации цикла и break – выход из цикла.

Задача:

Вычислить количество цифр целого числа (x).

Решение:

Идея заключается в том, чтобы последовательно «отбрасывать» цифру числа в разряде единиц. Для этого число надо нацело разделить на 10 и результат записать в исходное число. Отбросив цифру, ее необходимо одновременно посчитать, то есть увеличить значение счетчика (count) на 1. Эти два действия и будут повторяться некоторое количество раз, а значит, они составляют **тело** цикла.

Определим, до каких пор необходимо выполнять тело цикла? Если, например, исходное число x равно 24596, то количество повторений (итераций) будет равно 5:

Номер итерации	Значение переменной x	Значение переменной count
1	$x=24596/10=2459$	$count=0+1=1$
2	$x=2459/10=245$	$count=1+1=2$
3	$x=245/10=24$	$count=2+1=3$
4	$x=24/10=2$	$count=3+1=4$
5	$x=2/10=0$ цифры «закончились»	$count=4+1=5$

Таким образом, видим, что циклический процесс необходимо завершить, когда значение числа x стало равным нулю. Другими словами, пока $x>0$, будем выполнять тело цикла.

```
#include <stdio.h>
```

```
int main()
{
    int x;
    scanf("%d", &x);
    int count=0;
    while (x>0)
    {
        x/=10;
        count++;
    }
    printf("%d", count);
    return 0;
}
```

ОПЕРАТОР ЦИКЛА С ПОСТУСЛОВИЕМ DO .. WHILE

Общий вид записи:

do <тело_цикла> **while** (<выражение>;

Тело_цикла будет выполняться до тех пор, пока выражение истинно. Все, что говорилось выше справедливо и здесь, за исключением того, что данный цикл всегда выполняется хотя бы один раз, после чего проверяется, надо ли его выполнять еще раз.

ОПЕРАТОР ЦИКЛА FOR (С ПАРАМЕТРОМ)

Общий вид оператора:

for (<выражение1>; <выражение2>; <выражение3>) тело_цикла;

Цикл for эквивалентен последовательности инструкций:

```
<выражение1>;
while (<выражение2>)
{
    <тело_цикла>
    <выражение3>;
}
```

```
}
```

здесь выражение1 – инициализация счетчика (начальное значение), выражение2 – условие продолжения счета, выражение3 – увеличение счетчика. Выражения 1,2 и 3 могут отсутствовать (пустые выражения), но символы «;» опускать нельзя.

Например (1). для суммирования первых N натуральных чисел можно записать:

```
sum = 0;  
for (i=1; i<=N; i++) sum+=i;
```

Операция «запятая» чаще всего используется в операторе for. Она позволяет включать в его спецификацию несколько инициализирующих выражений. Предыдущий пример можно записать в виде:

```
for ( sum=0 , i=1; i<=N; sum+= i , i++);
```

ВЛОЖЕННЫЕ ЦИКЛЫ

Вложенным циклом называют конструкцию, в которой один цикл выполняется внутри другого. Внутренний цикл выполняется полностью во время каждой итерации внешнего цикла.

В программе можно использовать любые комбинации вложенных циклов всех типов: while, for, do..while.

Пример 2. Заполнение всего экрана символами '#'.

```
for (int i=1; i<=25; i++)  
    for (int j=1; j<=80; j++) cout<<'#';
```

В программе 25 раз осуществляется вывод по 80 символов.

УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ В ЦИКЛАХ

К операторам передачи управления относятся:

- оператор перехода к следующему шагу (итерации) цикла continue;
- выход из цикла break;
- оператор возврата из функции return (будет рассмотрен в соответствующей теме).

Оператор continue

Этот оператор может использоваться во всех типах циклов. Наличие оператора continue вызывает пропуск «оставшейся» части итерации и переход к началу следующей, т.е. досрочное завершение текущего шага и переход к следующему шагу.

В циклах while и do это означает непосредственный переход к проверочной части. В цикле for управление передается на шаг коррекции, т.е. модификации выражения_3.

Пример 3. Использование оператора continue

Фрагмент программы обработки только положительных чисел из общего потока значений задаваемых с клавиатуры, отрицательные значения пропускаются:

```
for (i=0; i<n; i++)
{
    scanf("%f", &a);
    if (a<0) continue;
    ... // обработка положительных элементов
}
```

Оператор continue часто используется, когда последующая часть цикла оказывается слишком сложной, так что рассмотрение условия, обратного проверяемому, приводит к слишком высокому уровню вложенности программы.

Оператор break

Оператор break производит экстренный выход из самого внутреннего цикла, к которому он принадлежит, и передает управление первому оператору, следующему за текущим оператором цикла.

Пример 4. Использование оператора break

Требуется определить задуманное число с 10 попыток.

...

```

int i=1;
//номер попытки имеет начальное значение, равное 1
while (i<=10)
{
    cin>>rez;
    if (rez==15) break;
    cout<<"\nПопытка неудачная.";
    i++;
}
if (i!=11) cout<<"\nВы угадали число.";
...

```

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Назовите 3 вида цикла.
2. Что такое тело цикла?
3. Какие три операции необходимо записать для организации любого вида цикла?
4. Пусть задан цикл с предусловием:

```

int n=5;
while (<выражение>)
{
    printf("%d",n);
    n--;
}

```

Каким должно быть логическое выражение в заголовке цикла, чтобы в результате его выполнения на экран были выведены:

- a. числа от 5 до 3?
 - b. числа от 5 до 1?
 - c. числа от 5 до 0?
5. Пусть задан цикл с предусловием:

```

int n=<значение>;
while (n<17)
{
    printf("%d",n);
}

```

```
<изменение значения n>;  
}
```

Каким должно быть начальное значение параметра цикла n и оператор изменения значения n , чтобы в результате его выполнения на экран были выведены:

- a. числа от 5 до 16 с шагом 1?
 - b. числа от -16 до 16 с шагом 2?
 - c. числа от 1 до 16, значения которых возрастают вдвое: 1, 2, 4, 8, 16?
6. Запишите цикл, полученный при выполнении задания 5а как цикл с параметром `for`.
 7. Запишите цикл, полученный при выполнении задания 5с как цикл с параметром `for`.
 8. Запишите цикл, полученный при выполнении задания 4b как цикл с параметром `for`.
 9. Запишите цикл, полученный при выполнении задания 4с как цикл с параметром `for`.

ПРАКТИЧЕСКАЯ РАБОТА. ИЗУЧЕНИЕ ОСОБЕННОСТЕЙ ОРГАНИЗАЦИИ ЦИКЛА С ПАРАМЕТРОМ (FOR)

Рассмотрим наиболее типичные варианты использования цикла с параметром `for`. Рекомендуется для каждого фрагмента (примера) составлять полноценную программу и тестировать ее работу при различных значениях параметров.

Пример 1. Счет в порядке убывания.

```
for (int i=5, r=1; i>=1; i--) r*=y;
```

Начальное значение $i=5$ будет при каждой итерации цикла **уменьшаться** на 1 за счет операции декремента ($i--$) до тех пор, пока не примет значение, равное 1 ($i>=1$).

Также стоит обратить внимание на начальную инициализацию переменной r ($r=1$) непосредственно в заголовке цикла.

Вопрос: вычисление какого значения реализует данный цикл?

Пример 2. Приращение при счете, отличное от 1 (единицы).

```
for (int n=5; n<61;n+=15) cout<<n;
```

Начальное значение $n=5$ при каждой итерации цикла будет **увеличиваться** на 15 за счет операции сложного присваивания ($n+=15$) до тех пор, пока не примет значение, равное 50 – последнее, не превышающее 61 ($n<61$).

Таким образом данный цикл организует вывод на экран значений: 5, 20, 35, 50.

Пример 3. Использование символов в качестве счетчика.

```
for (char ch='a'; ch<='z'; ch++) cout<<ch;
```

Данный цикл реализует задачу вывода на экран всех символов алфавита. Так как символы в таблице ASCII расположены последовательно, то операция инкремента ($ch++$) позволяет переходить к следующему по алфавиту символу.

Пример 4. Изменение счетчика в геометрической прогрессии.

```
for (int k=1; k<1000; k*=2) cout<<"Stepen="<<k;
```

В примере показаны возможности цикла for:

- проверять любое другое условие, отличное от числа итераций;
- задать возрастание значений счетчика по произвольной зависимости.

Итогом работы цикла будет вывод на экран степеней числа 2 ($k*=2$), не превышающих по значению 1000 ($k<1000$).

Пример 5. Неполный список выражений в заголовке цикла.

```
for (p=2; p<=202; ) p=p+n/k;
```

Изменение параметра цикла происходит не в заголовке, а в теле цикла.

Пример 6. Произвольное первое выражение в заголовке цикла.

```
for (cout<<"Вводите числа:"; p<=30;) cin>>p;
```

Первое выражение не обязательно должно инициализировать переменные, оно может быть любого типа. В этом фрагменте в качестве выражения1 используется оператор вывода cout. Сообщение о начале ввода данных выводится на печать один раз, а затем осуществляется прием вводимых чисел, пока не поступит число, превышающее 30 ($p<=30$).

Пример 7. Изменение управляющих переменных в теле цикла.

```
delta=12;  
for (n=1; n<1000; n+=delta)  
if (a>b) delta*=0.5;
```

Параметр delta изменен в процессе выполнения цикла.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Составить программу решения задачи в двух вариантах:

- используя цикл с параметром,
- используя цикл с предусловием.

1. Вычислить сумму первых N четных чисел.
2. Вычислить 10 -ую степень числа y .
3. Вывести на экран квадраты первых 25 чисел.
4. Составить программу, печатающую заглавные символы латинского алфавита и код каждого символа в десятичной и шестнадцатеричной системах счисления.
5. Даны целые числа K и N ($N > 0$). Вывести N раз число K .
6. Даны два целых числа A и B ($A < B$). Вывести в порядке возрастания все целые числа, расположенные между A и B (включая сами числа A и B), а также количество N этих чисел.
7. Даны два целых числа A и B ($A < B$). Найти сумму квадратов всех целых чисел от A до B включительно.
8. Вывести на экран таблицу умножения на 7.
9. Вычислить факториал числа N .
10. Дано целое число $N > 0$.
Найти сумму $N^2 + (N + 1)^2 + (N + 2)^2 + \dots + (2 \cdot N)^2$ (целое число).
11. Дано целое число $N > 0$. Найти произведение $1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$ (N сомножителей).
12. Дано целое число $N > 0$. Найти квадрат данного числа, используя для его вычисления следующую формулу: $N^2 = 1 + 3 + 5 + \dots + (2 \cdot N - 1)$. После добавления к сумме каждого слагаемого выводить текущее значение суммы (в результате будут выведены квадраты всех целых чисел от 1 до N).
13. Дано вещественное число A и целое число $N > 0$. Используя один цикл, вывести все целые степени числа A от 1 до N .
14. Вычислить сумму ряда $1+1.5+2+2.5+3+3.5+ \dots + 30$.

15. Составить программу вывода на экран «диалога» следующего вида:
 привет
 пока
 привет
 пока
 ...
 Всего строк должно быть N.

ДОПОЛНИТЕЛЬНЫЕ ЗАДАЧИ

1. Вычислить значение арифметического выражения:

$$R = \sqrt{98 + \sqrt{95 + \sqrt{92 + \dots + \sqrt{5 + \sqrt{2}}}}}$$

2. Вычислить значение дроби:

$$Q = \frac{1}{1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{\dots + \frac{1}{98 + \frac{1}{99 + \frac{1}{100}}}}}}}$$

3. Распечатать числовую последовательность, которая задается по следующим правилам: первое число последовательности – произвольное нечетное число от 3 до 99; каждый следующий элемент последовательности определяется через предыдущий элемент p , и равен

$3p+1$, если p нечётное число,

$p/2$, если p чётное число.

Например: 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Вычисления прекратить, когда очередной элемент последовательности станет равен 1. (Известно, что в любой такой последовательности рано или поздно встречается 1.)

Самодельный прибор для определения изменения погоды

Зная, как изменяется атмосферное давление, можно предположить, как изменится погода. Вы не задумывались о том, как измеряли атмосферное давление, не имея современных барометров?

В книге Жюль Верна «Дети капитана Гранта» есть интересный момент: «Штурмгласс показывал бурю». Штурмгласс (сосуд, содержащий жидкость сложного состава) использовали для предсказания изменений погоды. Изготовить такой «барометр» довольно сложно, так как он содержит специфические ингредиенты (камфору, калийную селитру). Но можно изготовить другие приборы, помогающие определить погоду.

Прежде чем заняться изготовлением такого прибора, рассмотрим историю создания и использования приборов, позволяющих определять изменения погоды.

Весь период наблюдений за метеорологическими элементами можно разделить на две неравные части: неинструментальные, визуальные наблюдения, и инструментальные.

Одним из первых метеорологических приборов был термометр, изобретенный Галилеем в 1597 году.

На парусных кораблях уже в XVII веке использовали прибор – штурмгласс, прогнозирующий погоду.

Торричелли навсегда вошел в историю как человек, доказавший существование такого понятия как атмосферное давление. Он впервые сформулировал гипотезу, что человек живет на дне воздушного океана, постоянно оказывающего на него определенное давление, и предложил одному из своих учеников измерить его величину при помощи запаянной трубки, наполненной ртутью.

Когда трубка опрокидывалась в сосуд с ртутью, вещество оставалось на определенной высоте, а над ртутью в трубке появлялось пустое пространство.

Именно в ходе данного эксперимента был сконструирован первый барометр. Этот опыт, проведенный Торричелли в 1643 году,



положил начало научной метеорологии. Ртутный барометр отличается очень высокой точностью измерения атмосферного давления и широко применяется в метеорологии.

Интересное описание барометра Вольфганга Гете, которое называли «погодное стекло». «Оно» было найдено после его смерти в его спальне в городе Веймаре в 1832 году. Сначала барометр приняли за декоративное настенное украшение, так как никто не мог понять назначение этого прибора. И только позднее в дневниках Гете нашлись заметки, из которых стало ясно, что это был измерительный прибор и для чего он был предназначен. Авторство изобретения было приписано В. Гете и получило его имя: «барометр Гете» или «погодное стекло Гете». Мало кому известно, что В. Гете кроме литературы занимался еще медициной и естествознанием. Из-под его пера вышли работы по астрономии и минералогии, он писал прекрасные картины.

Гете интересовался метеорологией. В его доме было несколько барометров. Об интересе, проявляемом Гете к этому прибору говорит и то, что он повесил его в своей спальне и наблюдал за его показаниями. Барометр представлял собой запаянный сверху стеклянный сосуд грушевидной формы. Сторона, прилегающая к стене, была плоской, из нижней части сосуда поднимался открытый сверху изящный изогнутый носик. С помощью своего барометра Гете следил за изменениями атмосферного давления. Из оставленных В. Гете бумаг стало известно, что однажды он в течение целого месяца записывал показания барометра в разных городах: в Веймаре и Йене, Лондоне, Бостоне и Вене и составлял сравнительные таблицы. Этот барометр по праву считают старейшим барометром в мире.

Особенной популярностью пользовались водяные барометры у моряков и жителей побережья, где климат не отличался постоянством. Сначала такие приборы выглядели как узкие стеклянные трубки-сосуды диаметром 4–6 см, затем нижняя часть большого сосуда постепенно расширялась и принимала форму груши, вместо обычной воды для большего эффекта стали брать подкрашенную воду, по бокам появились цветные стеклянные украшения. Вскоре из приборов, предназначенных для непосредственного измерения давления, эти барометры превратились еще и в произведения искусства. Их вешали на стены, ставили на столы, выполняли в виде шаров и т.д. Чего стоит, например, барометр-лебедь.



Конечно, такой прибор нельзя было назвать настоящим измерительным прибором, т.к. у него не было даже шкалы, но он достаточно хорошо показывал колебания атмосферного давления при перемене погоды. Однако, с 19 века, когда учеными были выявлены все его недостатки, как точного измерительного прибора, он потерял свое научное значение и остался в качестве декоративного украшения.

Это лишь краткая история появления и развития первых основных метеорологических приборов, в частности, барометра. На смену старым приборам пришли новые, позволяющие измерять метеорологические параметры автоматически. Однако и приборы, созданные в прошлом, до сих пор служат метеорологии.

Метеорологи много работают над совершенствованием приборов и аппаратов, действующих на принципах физики. Они широко используют компьютеры, применяют на спутниках сложную оптическую аппаратуру. И хотя мы часто слышим прогноз погоды, но нам иногда важно самим определить какая будет погода. Для этого можно самим изготовить приборы, позволяющие определить, ждёт ли дождь или нет.

Рассмотрим принцип действия барометра Гете.

Внешнее атмосферное давление воздуха действует на поверхность воды в открытом длинном носике и меняет ее уровень. Сначала, выбрав момент, когда атмосферное давление равно нормальному (760 мм рт.ст.) сосуд заполняется кипяченой или дистиллированной водой так, чтобы жидкость покрывала отверстие между носиком и большим сосудом. Тогда давление воздуха внутри большого сосуда будет равно внешнему нормальному атмосферному давлению. Чем больше объем воздуха заключен в верхней части большого сосуда, тем сильнее будут колебания уровня воды в носике при изменениях атмосферного давления. Установить барометр или повесить его надо в таком месте, где температура достаточно постоянна и сильно не меняется. Кроме того надо следить, чтобы вода не испарялась. И изменения температуры, и испарение воды будут искажать картину изменения атмосферного давления. Именно сильная зависимость показаний от колебаний температуры окружающей среды не позволила использовать барометр Гете для точных измерений. А поддерживать абсолютное постоянство температуры даже в домашних условиях невозможно. *При высоком атмосферном давлении (т.е. при хорошей погоде) уровень жидкости в носике понижается. При низком атмосферном давлении (при плохой погоде) уровень воды в носике повышается.*

Но все же для изготовления подобного прибора необходимо взять стеклянную бутылку от сока (можно пластиковую). Из изолянта приклеить на ней измерительную



шкалу. На трубочке, сделать пробку из изолянты. Она должна плотно входить в горлышко бутылки. Затем горлышко залить клеем, чтобы не проходил воздух. Бутылку на одну треть заполнить водой, лучше брать дистиллированную воду, поскольку обычная через год зацветает. Воду можно слегка подкрасить. Можно взять кипяченую воду и закрасить марганцовкой. Барометр готов.

Минус этого барометра в том, что при повышении температуры приводит к повышению жидкости через трубочку.

Таким образом, изменение уровня жидкости в трубке показывает изменение атмосферного давления. Если из трубки начнут выходить пузырьки воздуха, значит, давление очень высокое, а это к ясной устойчивой погоде; если вода начнет выливаться через верх трубки, давление низкое, можно ждать бури.

Еще один прибор для определения погоды можно изготовить из лампочки.

Необходимые материалы и инструменты для работы: перегоревшая лампочка, шуруповерт, вода, йод.

Принцип работы барометра: при появлении в атмосферном воздухе влаги больше, чем её содержится внутри колбы, она конденсируется (как роса) в пространстве выше налитой воды.

Взять перегоревшую электрическую лампочку. И там, где начинается цоколь с резьбовой частью, аккуратно просверлить небольшое отверстие. Сверлить нужно очень осторожно, прикладывая минимальное усилие.



Когда отверстие будет просверлено, залить в него подкрашенную водопроводную воду, заполнив стеклянную колбу до половины. Остается подвесить барометр между оконными рамами, с северной стороны, где на него не будут попадать прямые солнечные лучи.

Результат. По характеру конденсата можно судить о погоде:

1. Пространство в колбе выше воды чистое, без каких-либо капель – ожидается сухая, солнечная погода.
2. Мелкие частые капли конденсата – завтра будет облачно, но без осадков.
3. Капли средней величины, а между ними – сухие дорожки по вертикали – переменная облачность, где-то близко дождь, но кратковременный.
4. Редкие крупные капли – облачность, кратковременный дождь.
5. Все пространство, свободное от воды, в крупных каплях, которые, соединяясь между собой, стекают в воду – скоро будет сильный дождь.

6. На улице идёт дождь, а в колбе почти нет конденсата – вскоре дождь закончится.

7. Крупные капли только внизу, у поверхности воды в лампочке – дождь пройдёт стороной.

При минусовых температурах этот барометр, увы, не работает.

Попробуйте изготовить такие приборы, позволяющие определить будет ясно и солнечно или ждать осадки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Литература по математике:

1. Савенков, А.И. Содержание и организация исследовательского обучения школьников. М., 2003.
2. Выготский, Л.С. Проблема обучения и творческого развития в школьном возрасте // Избран, психол. исследования. М., 1982. Т.1. – С. 24.
3. Савенков, А.И. Исследовательское обучение и проектирование в современном образовании // Школьные технологии. М., 2004. № 4. С. 83–84.
4. Вединеева, Н.А. Развитие научно-исследовательской деятельности учителя и учащегося в школьной практике // Оренбург, 2004. № 3. С.6–7.
5. Сухомлинский, В.А. Павлышская средняя школа. М., 1979. – С. 106.
6. Райхмист, Р.Б. «Задачи по математике».
7. Студенецкая, В.Н. «Готовимся к ЕГЭ по математике».
8. Богомолов, Н.В. «Практические занятия по математике».
9. Письменный, Д.Т. «Готовимся к экзамену по математике».
10. http://www.softforfree.com/programs/rotation_bodies-29580.html
11. Старшинин, А.В. MicrosoftPowerPoint одним взглядом: Полный справочник по пакету подготовки презентаций и слайд-фильмов PowerPoint. – СПб.: ВHV-Санкт-Петербург, 1996. – 112 с.

Литература по информатике:

1. Бусько, В.Л. Основы алгоритмизации и программирования в среде Visual C++ : лаб. практикум по курсу «Основы алгоритмизации и программирования» для студ. 1–2-го курсов всех спец. БГУИР / В.Л. Бусько, А.А. Навроцкий. – Минск: БГУИР, 2008. – 66 с.: ил.
2. Дейтел, Х. Как программировать на C. 7-е издание. / Х. Дейтел, П. Дейтел. – М.: БИНОМ, 2017. – 1000 с.

3. Звягина, А.С. Основы программирования на языке Pascal: Лабораторный практикум. Ч.1 / А.С. Звягина. – Хабаровск, изд-во ДВГГУ, 2007. – 56 с.
4. Керниган, Б. Язык программирования С. Второе издание / Б. Керниган, Д. Ритчи – М.: «Вильямс», 2008. – 304 с.
5. Культин, Н. Б. С/С++ в задачах и примерах: 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2009. – 368 с.: ил. + CD-ROM
6. Поляков, К.Ю. Язык программирования Си. Практический курс. Режим доступа: <http://kpolyakov.narod.ru/school/c.htm>
7. Аверкин, В.П., Бобровский, А.И. и др. Программирование на С++ / под ред. проф. А.Д. Хомоненко – СПб.: БХВ-Петербург, 2009. – 249 с.

**Краевое государственное
бюджетное образовательное учреждение
дополнительного образования
«Хабаровский краевой центр развития творчества
детей и юношества»**

680000, г. Хабаровск, ул. Комсомольская, 87

тел. / факс: (4212) 30-57-13

Твиттер: twitter.com/yung_khb

Инстаграм: [@yung_khb](https://www.instagram.com/yung_khb)

e-mail: yung_khb@mail.ru

<http://www.kcdod.khb.ru>

Тираж: 18 экз.