

Министерство образования и науки Донецкой Народной Республики
Государственное профессиональное образовательное учреждение
«Донецкий политехнический колледж»

МЕТОДИЧЕСКАЯ РАЗРАБОТКА
открытого практического занятия

Тема: «**Программирование с использованием функций**»
ОП.05 «Основы программирования»
Специальность: 09.02.03 Программирование в компьютерных системах

Донецк, 2020

Методическая разработка открытого практического занятия по дисциплине ОП.05 «Основы программирования» для специальности 09.02.03 «Программирование в компьютерных системах» на тему «Программирование с использованием функций», 2020г.

Подготовила Мамедова Н.Д.. - преподаватель компьютерных дисциплин высшей квалификационной категории Государственного профессионального образовательного учреждения «Донецкий политехнический колледж», 2020.

Методическая разработка показывает возможности организации и проведения практического занятия по теме «Программирование с использованием функций» с применением интерактивных компьютерных технологий и прикладной программы Microsoft Visual Studio, используемой для разработки программ на языке программирования C++.

Рекомендована для преподавателей компьютерных дисциплин СПО

Рецензенты:

1. Червинский В.В. – к.т.н., доцент, доцент кафедры автоматизации и телекоммуникаций ГОУ ВПО «Донецкий национальный технический университет».
2. Кобцева Ю.В. – преподаватель высшей квалификационной категории ПЦК компьютерной техники ГПОУ «Донецкий политехнический колледж».

Рассмотрена и утверждена на заседании предметно-цикловой комиссии информатики и программирования
Протокол №3 от 08.10.2020г.

РЕЦЕНЗИЯ
на методическую разработку открытого занятия
по дисциплине ОП.05 Основы программирования
по теме «Программирование с использованием функций»
преподавателя ГПОУ «Донецкий политехнический колледж»
Мамедовой Н.Д.

Методическая разработка открытого занятия по теме «Программирование с использованием функций» составлена в соответствии с учебной программой дисциплины ОП.05 Основы программирования для специальности 09.02.03 Программирование в компьютерных системах. В ней отражена методика проведения практического занятия с использованием интерактивных форм обучения, реализации личностно-ориентированного подхода в обучении через применение информационных технологий, обеспечивающих интерактивное взаимодействие обучаемых и преподавателя на занятии.

Целью занятия по данной теме является овладение студентами практическими приемами разработки программ с использованием функций.

Дидактически обоснована форма проведения занятия и применение соответствующих ей методов обучения. Рационально сочетание методов коллективной и индивидуальной работы. Благодаря разнообразным формам организации фронтальной работы, эффективно решаются задачи, поставленные преподавателем. В течение занятия предполагается активное применение специального программного обеспечения – среды программирования Microsoft Visual Studio. Демонстрационные методы помогают лучшему восприятию, усвоению студентами соответствующего материала. Преподавателем организован деятельностный подход к изучению материала с учетом особенностей дисциплины «Основы программирования», предполагающих изучение указанной темы в ходе решения задач на компьютере. Это позволяет реализовать принцип связи теории с практикой, помогает студентам интегрировать знания из различных дисциплин, способствует развитию интеллектуальных умений студентов, воспитанию профессионально значимых качеств будущих специалистов.

Для самостоятельной работы студентов разработаны задания разного уровня сложности, умело подобраны задачи из курса математики, физики, экономики.

Для оценивания уровня знаний обучаемых эффективно используются возможности программной тестовой оболочки Конструктор тестов компании Keepsoft.

В разработке освещается прикладной характер дисциплины, отражена связь с другими дисциплинами учебного плана, подчеркивается важность практической составляющей для формирования ключевых профессиональных компетенций студентов специальности 09.02.03 Программирование в компьютерных системах.

Такая форма организации занятий способствует активизации познавательной деятельности студентов, реализации личностно-ориентированной технологии обучения, позволяет обеспечить гибкость процесса обучения.

Разработка оформлена в соответствии с существующими требованиями и полностью освещает выбранную тему, реализует цели, поставленные преподавателем.

Рецензент

к.т.н., доц., доц. кафедры автоматизации и телекоммуникаций
ГОУ ВПО «Донецкий национальный технический университет»

ПОДПИСЬ УДОСТОВЕРЯЮЩАЯ

Начальник ОК



В.В. Червинский

Н.Д. Мамедова

РЕЦЕНЗИЯ

на методическую разработку занятия по дисциплине ОП.05 Основы программирования по теме «Программирование с использованием функций» преподавателя Государственного профессионального образовательного учреждения «Донецкий политехнический колледж» Мамедовой Н.Д.

Данная методическая разработка открытого занятия посвящена теме «Программирование с использованием функций», составлена в соответствии с Государственным образовательным стандартом среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах. Целью разработки является демонстрация возможности проведения практического занятия с использованием интерактивных форм обучения с применением мультимедийных средств с учетом личностно-ориентированного подхода в обучении студентов.

В разработке освещается прикладной характер дисциплины, связь с другими дисциплинами учебного плана, подчеркивается важность практической составляющей в учебно-воспитательном процессе для формирования ключевых профессиональных компетенций студентов специальности 09.02.03 «Программирование в компьютерных системах». Благодаря умелому подбору задач по соответствующей теме и применённому методу открытых задач наглядно демонстрируются приемы алгоритмизации такой алгоритмической конструкции как вспомогательные алгоритмы и их программной реализации средствами языка C++ посредством использования функций. Несмотря на сложность, материал изложен в доступной понятной форме, качественно подобран иллюстративный материал. Для повышения эффективности усвоения данного материала на всех этапах занятия предусмотрено применение компьютерных технологий. Умело организовано повторение и закрепление изученного с помощью использования интерактивного теста с различными типами вопросов.

На протяжении занятия используются разнообразные формы организации практической деятельности студентов: самостоятельная, коллективная. Применены демонстрационно-иллюстративный, проблемный методы обучения.

Такое занятие позволяет преподавателю активизировать познавательную деятельность студентов, развивать интерес к выбранной специальности, умение творческого выполнения профессиональных задач, помогает студентам развивать алгоритмический стиль мышления, системно-комбинаторное мышление.

При создании методической разработки преподавателем была использована разнообразная методическая и научная база в виде учебников, пособий, а также видеоуроков и презентаций.

Разработка оформлена в соответствии с Методическими рекомендациями по написанию и оформлению методических материалов, разработанных Учебно-методическим центром профессионально-технического образования.

Рецензент:

Кобцева Ю.В., преподаватель высшей
квалификационной категории ПЦК
компьютерной техники ГПОУ
«Донецкий политехнический колледж»



СОДЕРЖАНИЕ

Введение.....	4
План занятия.....	6
Структура занятия.....	10
Ход занятия.....	11
1. Организационный момент.....	11
2. Ознакомление студентов с темой и целями занятия.....	11
3. Мотивация обучения.....	11
4. Проверка домашнего задания.....	12
5. Актуализация опорных знаний.....	16
6. Объяснение нового материала, коллективное выполнение практического задания.....	16
7. Закрепление нового материала, выполнение самостоятельной работы студентами.....	23
8. Итоги занятия.....	24
9. Домашнее задание.....	24
Приложение 1	25
Приложение 2.....	26
Приложение 3.....	27
Приложение 4.....	30
Приложение 5.....	39
Приложение 6.....	47
Приложение 7.....	48

Введение

Дисциплина «Основы программирования» относится к общепрофессиональным дисциплинам в структуре основной профессиональной образовательной программы подготовки будущих специалистов среднего звена по специальности 09.02.03 Программирование в компьютерных системах. Наиболее эффективными в процессе подготовки будущих техников-программистов являются практические занятия. Практические занятия, как виды учебных занятий, направлены на формирование учебных и профессиональных практических умений и составляют важную часть теоретической и профессиональной практической подготовки студентов. Именно на практических занятиях по программированию развиваются интеллектуальные умения студентов, формируется культура умственного труда.

Основная дидактическая цель практического занятия по теме «Программирование с использованием функций» – ознакомление с методикой решения задач с применением функций. В ходе работы у студентов формируются умения анализировать, делать выводы и обобщения, на основе анализа проектировать свою деятельность, намечать конкретные пути решения той или иной практической задачи, самостоятельно принимать решения аргументировано доказывать правильность своего решения, оформлять результаты своей работы и т.д.

Основным критерием, основанием для организации дифференцированной работы на занятии является достижение уровня обязательной подготовки каждого студента по определенной теме. Для организации практического занятия по теме «Программирование с использованием функций» подобран соответствующий дидактический материал, подготовлены разноуровневые задания. Студентам на этапе подготовки к данному занятию предлагаются теоретические вопросы, комплект заданий для самоподготовки, рекомендуемая литература и другие источники информации.

Демонстрация приемов и способов выполнения практических действий, использование типовых задач с постепенным усложнением структуры алгоритмов, использование «демонстрационных» программ поддерживают стиль «обучение на примерах», позволяют усваивать материал намного эффективнее, концентрируют внимание на принципиально важных приемах программирования. Решение задач прикладной направленности, создание проблемных ситуаций по теме занятия способствуют активизации мышления студентов. Организация контроля осуществляется с помощью компьютерного тестирования. В разработанном тесте по теме «Функции в C++» содержатся разные типы вопросов, что также дает возможность индивидуализировать и дифференцировать задания. Использование тестовой программной

оболочки позволяет студентам проанализировать ошибки и вернуться к неотработанным вопросам.

Индивидуальный контроль за качеством усвоения студентами учебного материала осуществляется в ходе самостоятельной работы студентов за компьютерами с использованием среды программирования. Для выполнения самостоятельной практической работы за компьютером подобраны задания из курса математических дисциплин, физики, экономики. Для каждого студента практические задания представлены тремя уровнями сложности. У обучаемых имеется возможность самостоятельного выбора уровня решаемых задач.

При подведении итогов занятия для оценки эффективности работы группы, увлекательности и полезности выбранных форм работы применяется рефлексия.

Материал методической разработки отображает содержание открытого занятия по дисциплине ОП.05 «Основы программирования» по теме «Программирование с использованием функций». Методическая разработка составлена на основе учебного плана специальности 09.02.03 «Программирование в компьютерных системах» ГПОУ «Донецкий политехнический колледж» и в соответствии с рабочей программой дисциплины.

В методической разработке предложено проведение практического занятия с использованием интерактивных форм обучения с применением мультимедийных средств с учетом личностно-ориентированного подхода в обучении студентов.

Методическая разработка рекомендована для использования в практической деятельности для преподавателей СПО.

ПЛАН ЗАНЯТИЯ

Дисциплина: ОП.05 «Основы программирования»

Специальность: 09.02.03 «Программирование в компьютерных системах»

Тема занятия: Программирование с использованием функций

Цели занятия:

Методическая:

- усовершенствовать и продемонстрировать методику проведения практического занятия с использованием интерактивных форм обучения, мультимедийных средств и прикладной программы Microsoft Visual Studio, используемой для разработки программ на языке программирования C++, с учётом личностно-ориентированного подхода в обучении.

Дидактическая (учебная):

- изучить особенности проектирования и реализации программ с функциями с использованием специализированных инструментальных средств;
- формировать практические умения описания и определения функций, использования механизма передачи параметров в функцию и возврата результатов из нее;
- закреплять умения применять теоретические знания при решении практических задач с использованием функций;
- формировать навыки самостоятельной работы студентов при выполнении поставленных задач, применять полученные знания по проектированию, разработке, тестированию программ с использованием функций.

Дидактическая (развивающая):

- развивать алгоритмический стиль мышления, критическое мышление, умение анализировать проектировать, планировать, умение связывать полученные теоретические знания с практикой;
- развивать навыки проектно-исследовательской деятельности, творческое отношение к выполнению профессионального задания, умение оптимально использовать возможности среды программирования в профессиональной деятельности.

Воспитательная:

- способствовать формированию эмоционально-ценностного отношения к практической деятельности, воспитанию творческой активности;
- способствовать формированию познавательного интереса, развития общих и профессиональных компетенций в процессе обучения;
- воспитывать самодисциплину, внимание, трудолюбие,

ответственность за результаты своей работы;

- формировать навыки самоконтроля и взаимоконтроля, формировать культуру коллективного труда, развивать культуру умственного труда.

Формируемые общие компетенции:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Решать проблемы, оценивать риски и принимать решения в нестандартных ситуациях.

ОК 4. Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности.

ОК 6. Работать в коллективе и команде, обеспечивать её сплочение, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Ставить цели, мотивировать деятельность подчинённых, организовывать и контролировать их работу с принятием на себя ответственности за результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

Формируемые профессиональные компетенции:

ПК 1.1. Выполнять разработку спецификаций отдельных компонент.

ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять оптимизацию программного кода модуля.

Тип занятия: занятие комплексного применения знаний и умений.

Вид занятия: практическая работа с изучением, углублением и обобщением знаний.

Методы и формы проведения занятия: объяснительно-иллюстративный метод с использованием мультимедийных технологий; использование активных методов обучения в форме фронтального опроса, компьютерного тестирования, индивидуальная работа на составление программ с учетом личностно-ориентированного подхода к студентам

Междисциплинарные связи:

Обеспечивающие: ОДп.01 Математика, ОДп.02 Информатика и ИКТ, ЕН.02 Элементы математической логики, ОП.08 Теория алгоритмов

Обеспечиваемые: МДК.01.01 Системное программирование, МДК.01.02 Прикладное программирование, МДК.01.03 Объектно-ориентированное программирование, УП.01 Учебная практика по модулю ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем, МДК.03.01 Технология разработки программного обеспечения, МДК.03.02 Инструментальные средства разработки программного обеспечения, ЕН.04 Численные методы в программировании

Методическое обеспечение:

- рабочая программа;
- план занятия;
- методическая разработка к занятию;
- презентация к занятию;
- инструкция выполнения практического задания (приложение 3);
- тест (приложение 4);
- раздаточный материал (карточки, словарь терминов) (приложение 1);
- набор прикладных программ MS Office: Word, PowerPoint;. Microsoft Visual Studio, тестовая оболочка для конструирования тестов «Конструктор тестов компании Keepsoft»

Технические средства обучения: интерактивная доска, мультимедийный проектор, ПК.

Продолжительность занятия – 80 минут

Литература**Основная:**

1. Каширин И. Ю., Новичков В. С. От С к С++: Учебное пособие для вузов. – 2 -е изд., стереотип. – М.: Горячая линия – Телеком, 2012. – 334 с.: ил.
2. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие/ Под ред. проф. Л.Г.Гагариной. – М.: ИД «ФОРУМ»: ИНФРА-М, 2006. – 416 с.: ил. – (Профессиональное образование).
3. Прата, Стивен. Язык программирования С++. Лекции и упражнения, 6-е изд. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2012. - 1248 с. : ил. - Парал. тит. англ.
4. Сабуров С.В.Языки программирования С и С++. М.: Бук-пресс, 2006. 647 с. (Справочное руководство пользователя персонального компьютера).

5. Бьерн Страуструп. Язык программирования C++. Специальное издание. Пер. с англ. — М.: Издательство Бином, 2011 г. — 1136 с: ил.
6. Т.А.Павловская С/C++. Программирование на языке высокого уровня, Питер, СПб, 2005. - 460 с.

Дополнительная:

1. Культин Н.Б. С\C++ в задачах и примерах: 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2009. – 368 с.: ил. +CD-ROM.
2. Т.А.Павловская, Ю.А.Щупак С/C++. Структурное программирование. Практикум, Питер 2005. – 238 с.
3. Е.Л.Романов Практикум по программированию на C++, БХВ-Петербург, СПб, 2004. – 427 с.
4. Л.М.Климова С++. Практическое программирование. Решение типовых задач. Кудиц-образ, 2001.- 596 с.

Интернет-ресурсы:

1. Интернет-Университет информационных технологий. Библиотека учебных курсов [Электронный ресурс]. - Режим доступа: <http://old.intuit.ru/courses.html>, свободный.
2. Библиотека электронных книг по изучению языка программирования С/C++[Электронный ресурс]. - Режим доступа: <http://netcode.ru/cpp/>свободный
3. Видео и онлайн-курсы по C++ для разработчиков разных уровней Microsoft [Электронный ресурс]. - Режим доступа: <https://proglib.io/p/cpp-video-and-online-courses/>, свободный

СТРУКТУРА ЗАНЯТИЯ

1.Организационный момент	1мин
- приветствие студентов;	
- отметка в журнале отсутствующих студентов;	
- проверка готовности к занятию студентов, аудитории, оборудования.	
2. Ознакомление студентов с темой и целями занятия	1 мин
3.Мотивация обучения	2мин
4.Проверка домашнего задания:	10 мин
-устный опрос по теме «Функции»	
-проверка решения задач	
5. Актуализация опорных знаний в форме компьютерного тестирования	8 мин
6. Объяснение нового материала, коллективное выполнение практического задания	35 мин
7. Закрепление нового материала, выполнение самостоятельной работы студентами	20мин
8. Подведение итогов занятия:	2мин
-анализ работы на уроке;	
- выставление оценок;	
-рефлексия	
9. Домашнее задание	1мин

ХОД ЗАНЯТИЯ

1. Организационный момент

- приветствие студентов;
- отметка в журнале отсутствующих студентов;
- проверка готовности к занятию студентов, аудитории, оборудования.

2. Ознакомление студентов с темой и целями занятия.

Тема сегодняшнего практического занятия – «Программирование с использованием функций».

Преподаватель объявляет цели занятия:

- 1) научиться самостоятельно проектировать программы модульной структуры;
- 2) закрепить умения в применении теоретических знаний по теме «Использование функций» при решении практических задач,
- 3) сформировать навыки самостоятельной работы за компьютером при выполнении поставленной задачи с использованием среды программирования.

3. Мотивация обучения.

Задачи, которые встречаются в практике программирования, зачастую имеют сложную логическую структуру или громоздкий алгоритм решения. Для упрощения восприятия таких алгоритмов их разделяют на подзадачи, каждая из которых рассматривается как отдельная задача и может быть решена для заданных значений ее аргументов. Метод последовательной детализации позволяет составить алгоритм из действий, которые, не являясь простыми, сами представляют собой достаточно самостоятельные алгоритмы. В этом случае говорят о вспомогательных алгоритмах или подпрограммах (функциях).

Каковы же преимущества использования функций в программах?

Разделение решения задачи на отдельные функции позволяет некоторым из них тоже использовать при решении других задач и даже другими пользователями. Использование подпрограмм позволяет избежать избыточности кода, структурировать процесс отладки, создавать библиотеки подпрограмм, которые чаще всего используются. Итак, функции позволяют сделать программу модульной, то есть разделить программу на несколько маленьких подпрограмм (функций), которые в совокупности выполняют поставленную задачу. Еще один огромный плюс функций в том, что их можно многократно использовать. Данная возможность позволяет многократно использовать один раз написанный код, что в свою очередь, намного сокращает объем кода программы! Использование функций является первым шагом к повышению степени абстракции программы и ведет к упрощению ее структуры.

“Модульность – фундаментальный аспект всех успешно работающих больших систем” (Б. Страуструп).

Существенная часть умения программирования — это умение проектировать программный продукт, в частности, разбивать задачу на подзадачи.

Использование функций в программах дает следующие преимущества:

- компактная организация программы путем удобного вызова программного кода по его имени, который в программе может встречаться несколько раз (повторяться);
- экономия памяти, размера исходного и исполнительного кода и т.д.;
- уменьшение риска возникновения ошибок для больших наборов кодов;
- повышение читабельности программного кода.

Сегодняшняя тема имеет большое прикладное значение. Проектирование программ модульной структуры пригодится как и при изучении данной дисциплины, так и при изучении дисциплин Системное программирование, Прикладное программирование, других специальных дисциплин.

4. Проверка домашнего задания

Домашнее задание состоит в подготовке ответов на вопросы к материалам по теме «Функции» и решении задач по данной теме.

4.1 Фронтальный устный опрос по теме «Функции в C++»

Преподаватель выборочно задает вопросы для устного опроса (приложение 2).

Преподаватель: Дать определение функции. Для чего функции используются в программах?

Предполагаемый ответ студента: Функция — это поименованный набор описаний и операторов, выполняющих определенную задачу. Использование подпрограмм позволяет избежать избыточности кода, структурировать процесс отладки, создавать библиотеки подпрограмм, которые чаще всего используются.

Преподаватель: Что такое определение функции?

Предполагаемый ответ студента: определение функции - определение функции, включающее заголовок функции (тип возвращаемого значения, имя функции, типы и число формальных параметров) и тело функции.

Преподаватель: Для чего нужен прототип? Всегда ли его присутствие необходимо в программе, содержащей функции?

Предполагаемый ответ студента: Прототип функции - объявление функции, не содержащее тела функции, но указывающее имя функции, количество, типы аргументов и возвращаемый тип данных в

качестве результата. Нет. Если вызываемая функция описана раньше, чем функция её вызывающая, то прототип не обязателен.

Преподаватель: Признаком чего служит слово `void`, стоящее перед именем функции?

Предполагаемый ответ студента: Слово `void` – признак нетипизированной функции, которая не возвращает значений.

Преподаватель: Признаком чего служит слово `void`, стоящее на месте параметров? Всегда ли оно необходимо?

Предполагаемый ответ студента: Слово `void`, стоящее на месте параметров означает отсутствие параметров. Не всегда.

Преподаватель: Как осуществляется выход из функции?

Предполагаемый ответ студента: С помощью оператора `return` или по достижении конца функции.

Преподаватель: В чем разница между функциями, не возвращающими никаких значений и возвращающими значения объявленного типа?

Предполагаемый ответ студента: Типизированная функция всегда возвращает одно обязательное значение, вызывается в выражении. Нетипизированная не возвращает значения через имя функции, её вызов представляет собой самостоятельную инструкцию.

Преподаватель: Что называют формальными параметрами функции и что фактическими? Как эти два понятия соотносятся между собой?

Предполагаемый ответ студента: Фактический параметр — аргумент, передаваемый в функцию при ее вызове. Формальный параметр — аргумент, указываемый при объявлении или определении функции. Между ними должно соблюдаться соответствие (количество, порядок следования, совместимость по типу).

Преподаватель: Какие переменные называются локальными и какие глобальными?

Предполагаемый ответ студента: локальная переменная — переменная, описанная внутри тела функции (или аргумент в списке формальных параметров), глобальная переменная — переменная, описанная вне всех функций.

Преподаватель: В чем заключается особенность передачи массивов в качестве параметров функций?

Предполагаемый ответ студента: Для передачи массива чисел в функцию, нужно передать указатель на этот массив чисел. Указателем на массив чисел является имя массива или адрес первого элемента массива.

Преподаватель: Можно ли возвращать массив из функции, пользуясь оператором `return`?

Предполагаемый ответ студента: Нет.

Комментирование ответов студентов.

4.2 Проверка решения задач:

Демонстрация решения задач прикладного характера (просмотр видеоролика с объяснением решения задач прикладного характера). Осуществление самопроверки. Выборочная проверка решений задач у отдельных студентов. Анализ ошибок, допущенных при решении задач. Задания для самостоятельной подготовки приведены в приложении 3.

Задача 1. Составим программу с использованием подпрограммы (функции). Опишем функцию, которая находит длину произвольного отрезка, концы которого заданы координатами. Применим формулу из геометрии

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
#include <iostream>
using namespace std;
```

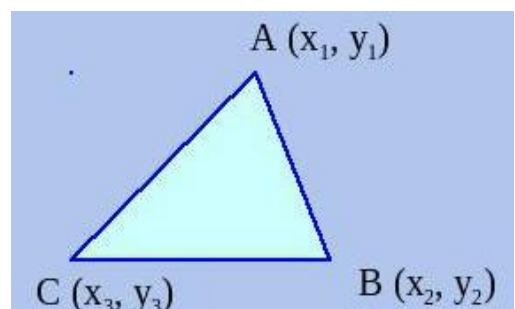
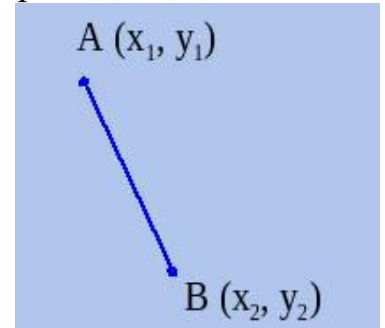
```
double length (double x1, double y1, double x2, double y2)
{
    double len;
    len = sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
    return len;
}

int main()
{
    double x1, y1, x2, y2;
    double dlina; // результат
    cout << "Enter data: " << endl;
    cout << "x1 = "; cin >> x1;
    cout << "y1 = "; cin >> y1;
    cout << "x2 = "; cin >> x2;
    cout << "y2 = "; cin >> y2;

    dlina = Length(x1, y1, x2, y2); // вызов функции

    cout << "length = " << dlina << endl;
    return 0;
}
```

Задача 2. Составить программу вычисления периметра треугольника, если известны координаты его вершин. Для вычисления длины отрезка будем использовать функцию length(). Чтобы вычислить периметр, вызовем её три раза для определения длин трех сторон треугольника.



```

#include <iostream>
using namespace std;
double length (double, double, double, double);

int main()
{
    double x1, y1, x2, y2, x3, y3;
    double dlina1, dlina2, dlina3;

    cout << "Enter data: " << endl;
    cout << "x1 = "; cin >> x1;
    cout << "y1 = "; cin >> y1;
    cout << "x2 = "; cin >> x2;
    cout << "y2 = "; cin >> y2;
    cout << "x3 = "; cin >> x3;
    cout << "y3 = "; cin >> y3;

    dlina1 = Length(x1, y1, x2, y2); // вызов функции
    dlina2 = Length(x2, y2, x3, y3); // вызов функции
    dlina3 = Length(x3, y3, x1, y1); // вызов функции

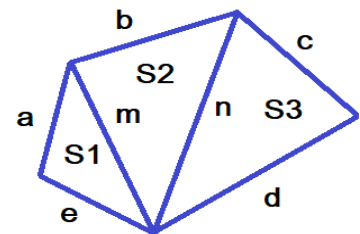
    double P= dlina1+ dlina2+ dlina3; // вычисление периметра

    cout << "Perimeter = " << P << endl;
    return 0;
}

double length (double x1, double y1, double x2, double y2)
{
    double len;
    len = sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
    return len;
}

```

Задача 3. Составить программу вычисления площади пятиугольника со сторонами a , b , c , d , e и диагоналями m , n , используя функцию вычисления площади произвольного треугольника по формуле Герона



$$S = S1 + S2 + S3$$

```

#include <iostream>
#include <math.h>
using namespace std;
double Geron (double a, double b, double c)
{

```

```

double p=(a+b+c)/2;
double S;
S = sqrt(p*(p-a)*(p-b)*(b-c));
return S;
}

int main()
{
double a,b,c,d,e,m,n;
double S,S1,S2,S3; // результат
cout << "Enter data: " << endl;
cout << "a = "; cin >> a;
cout << "b = "; cin >> b;
cout << "c = "; cin >> c;
cout << "d = "; cin >> d;
cout << "e = "; cin >> e;
cout << "m = "; cin >> m;
cout << "n = "; cin >> n;
S1= Geron(a,m,e); // вызов функции
S2= Geron(b,m,n); // вызов функции
S3= Geron(c,n,d); // вызов функции
S=S1+S2+S3;
cout << "Square = " << S<< endl;
return 0;
}

```

5. Актуализация опорных знаний

Актуализация опорных знаний проводится в форме компьютерного тестирования (приложение 4). Тестирование выполняется индивидуально каждым студентом. Затем осуществляется коллективная проверка результатов теста на интерактивной доске. Комментирование результатов. Похвала студентов, получивших максимальные баллы за тест.

6. Объяснение нового материала, коллективное выполнение практического задания

Задания выполняются преподавателем и студентами с комментариями и подсказками на интерактивной доске, студенты компилируют полученные решения, работая за компьютером и используя среду программирования Microsoft Visual Studio). Преподаватель следит за работой всех студентов, контролирует полученные результаты работы, оказывает помощь наиболее отстающим.

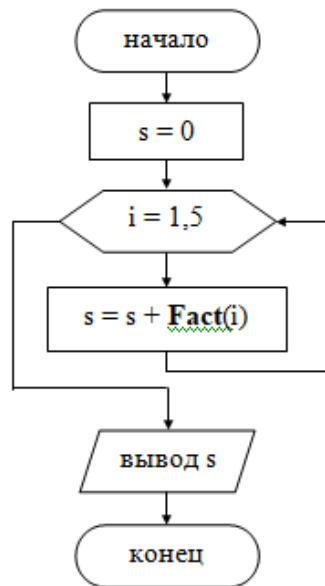
Преподаватель перед началом работы за компьютером проводит инструктаж по технике безопасности и правилам работы за ПК.

Коллективное решение задач прикладного характера.

Задача 1. Вычислить сумму $1!+2!+3!+\dots+5!$.

Блок-схема:

Функция main()



Функция Fact()

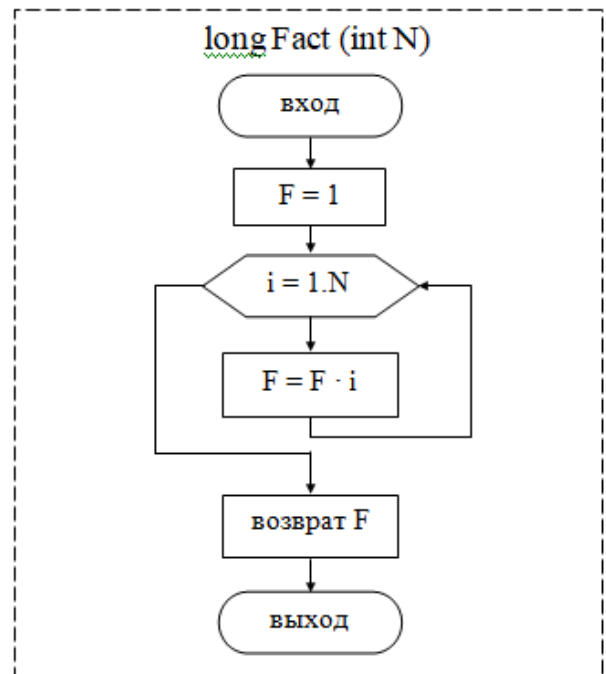


Рисунок 1. Блок-схема решения задачи 1

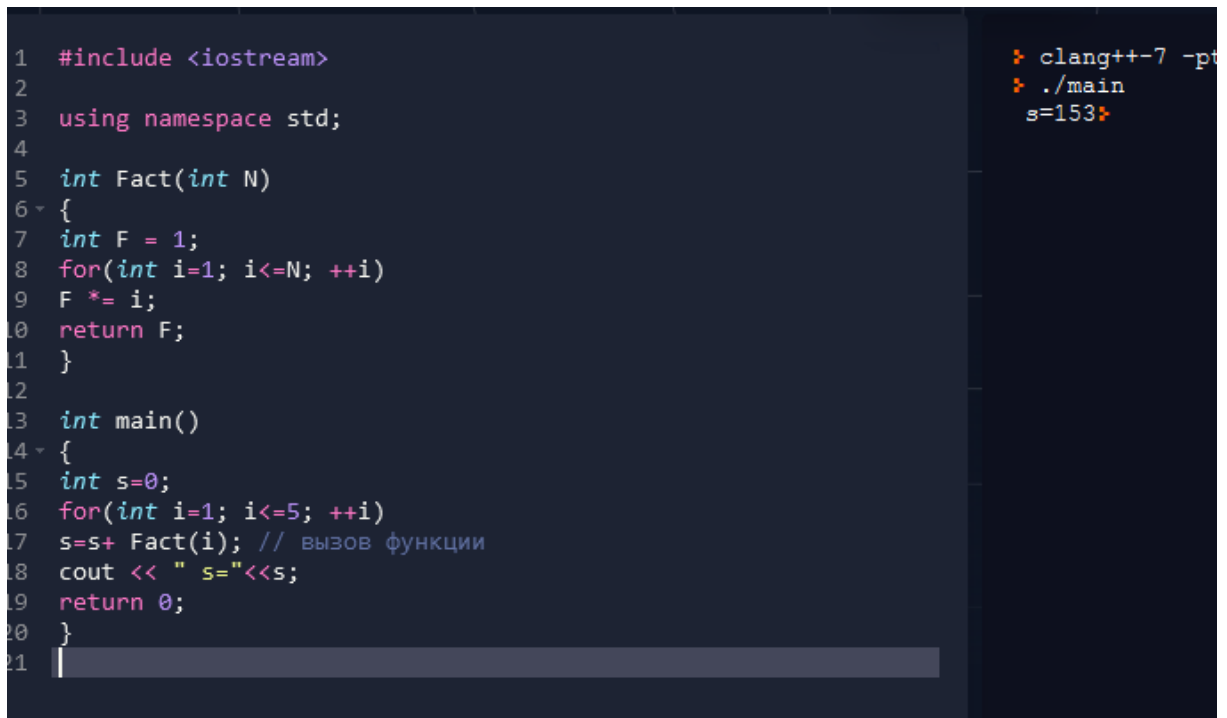
```
#include <iostream>
```

```
using namespace std;
```

```
long Fact(int N)
{
    long F = 1;
    for(int i=1; i<=N; ++i)
        F *= i;
    return F;
}
```

```
int main()
{
    long s=0;
    for(int i=1; i<=5; ++i)
        s=s+ Fact(i); // вызов функции
    cout << " s="<<s;
    return 0;
}
```

Действительно получили правильный результат :
 $1!+2!+3!+4!+5!=1+2+6+24+120=153$



```
1  #include <iostream>
2
3  using namespace std;
4
5  int Fact(int N)
6  {
7      int F = 1;
8      for(int i=1; i<=N; ++i)
9          F *= i;
10     return F;
11 }
12
13 int main()
14 {
15     int s=0;
16     for(int i=1; i<=5; ++i)
17         s=s+ Fact(i); // вызов функции
18     cout << " s="<<s;
19     return 0;
20 }
21 |
```

clang++-7 -pt
./main
s=153

Рисунок 2. Результаты решения задачи 1

Задача 2. Составьте программу вычисления числа сочетаний из N по M.

Число сочетаний определяется по формуле: $C_N^M = \frac{N!}{M!(N-M)!}$, где $n!=1\cdot2\cdot3\cdot\ldots\cdot n$. Вычисление факториала оформить в виде подпрограммы-функции.

При решении задачи 2 воспользуемся разработанной ранее функцией для вычисления факториала Fact().

```
#include <iostream>
using namespace std;
int Fact(int N)
{
    int F = 1;
    for(int i=1; i<=N; ++i)
        F *= i;
    return F;
}
int main()
{
    int CMN, N, M;
    cout<<"\n Введите N, M (N>=M)";
    cin>>N>>M;
    CMN= Fact(N)/ (Fact(M) *Fact(N-M));// вызов функции
```

```
cout<<"\n CMN="<<CMN;
return 0;
}
```

```
Введите N, М (N>=М) 5 3

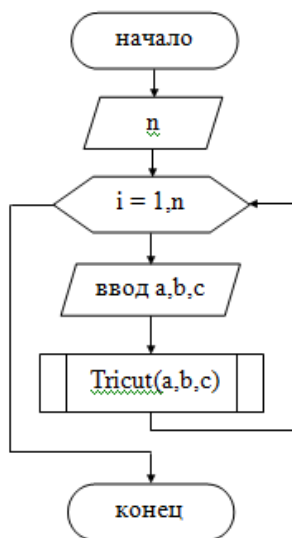
CMN=10
```

Рисунок 3. Результаты решения задачи 2

Задача 3. Составить функцию вычисления площади и высот треугольника со сторонами a , b , c по формулам $h_a = \frac{2S}{a}$, $h_b = \frac{2S}{b}$, $h_c = \frac{2S}{c}$, где $S = \sqrt{p(p-a)(p-b)(p-c)}$, $p = \frac{a+b+c}{2}$, если заданы стороны треугольника a, b, c . Вычислить площадь и высоты нескольких треугольников. Предусмотреть в функции проверку возможности существования треугольника для вводимых длин сторон.

Блок-схема

Функция `main()`



Функция

`void Tricut(double x, double y, double z)`

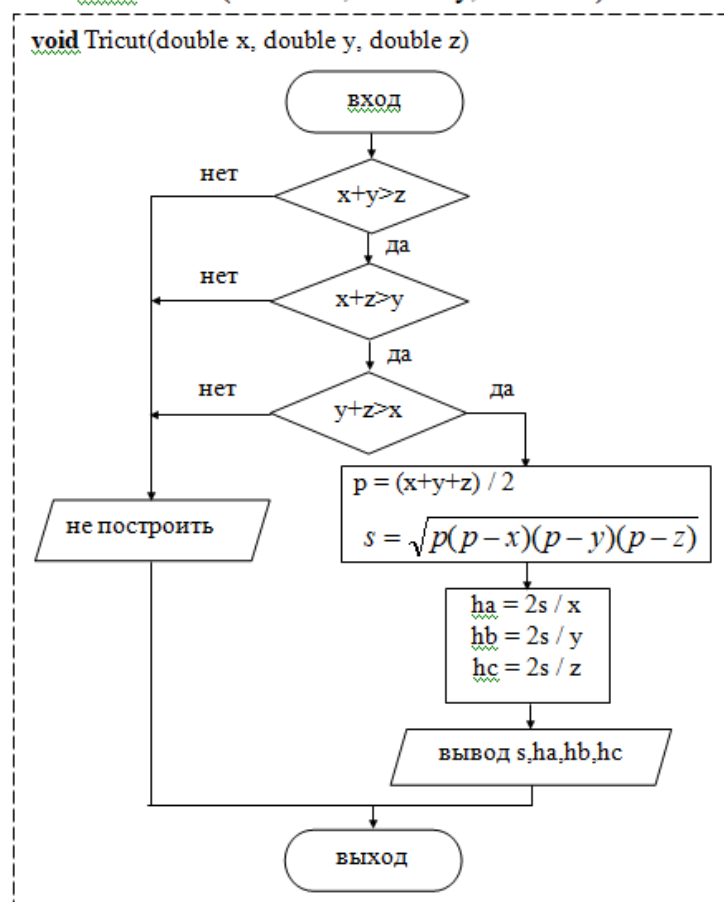


Рисунок 4. Блок-схема решения задачи 3

```

#include <iostream>
#include <math.h>
using namespace std;

void Tricut(double x,double y, double z)
{ double p, S, ha, hb, hc;
if (x+y>z && x+z>y&& y+z>x)
    {p=(x+y+z)/2;
    S=sqrt(p*(p-x)*(p-y)*(p-z));
    ha=2*S/x;
    hb=2*S/y;
    hc=2*S/z;
    cout << "\n S="<<S;
    cout << "\n ha="<<ha;
    cout << "\n hb="<<hb;
    cout << "\n hc="<<hc;
    }
    else cout << " \n Не существует треугольник!!!";
}

int main()
{
int n; double a,b,c;
cout<<"\n Введите n:";
cin>>n;
for(int i=1; i<=n; i++) {
    cout<<"\n Введите стороны: ";
    cin>>a>>b>>c;
    Tricut(a,b,c);}
return 0;
}

```

```

clang++-7 -pthread -std=c++17 -o main main.cpp
./main

Введите n:3

Введите стороны: 3 4 5

S=6
ha=4
hb=3
hc=2.4
Введите стороны: 4.56 6.7 8.45

S=15.2089
ha=6.67059
hb=4.53998
hc=3.59975
Введите стороны: 3 5 10

Не существует треугольник!!!

```

Рисунок 5. Результаты решения задачи 3

Решение задачи повышенной сложности.

Задача на использование ссылок в качестве параметров, использование одномерного массива в качестве аргумента функции.

Задача 4.

Составить функцию FnMax определения максимума функции $y = f(x)$, заданной в дискретных точках отрезка $[a, b]$ с постоянным шагом h . Используя её, найти максимумы функций: $y_1 = e^{-x} \cos^2 x - 1$, $y_2 = x \ln|x+1|$.

Значения функций в дискретных точках сохраняется в массиве $z = (z_1, z_2, \dots, z_n)$, среди значений массива определяется максимальное значение.

Схема нисходящего проектирования алгоритма решения задачи в виде структурограммы представлена на рисунке 6.

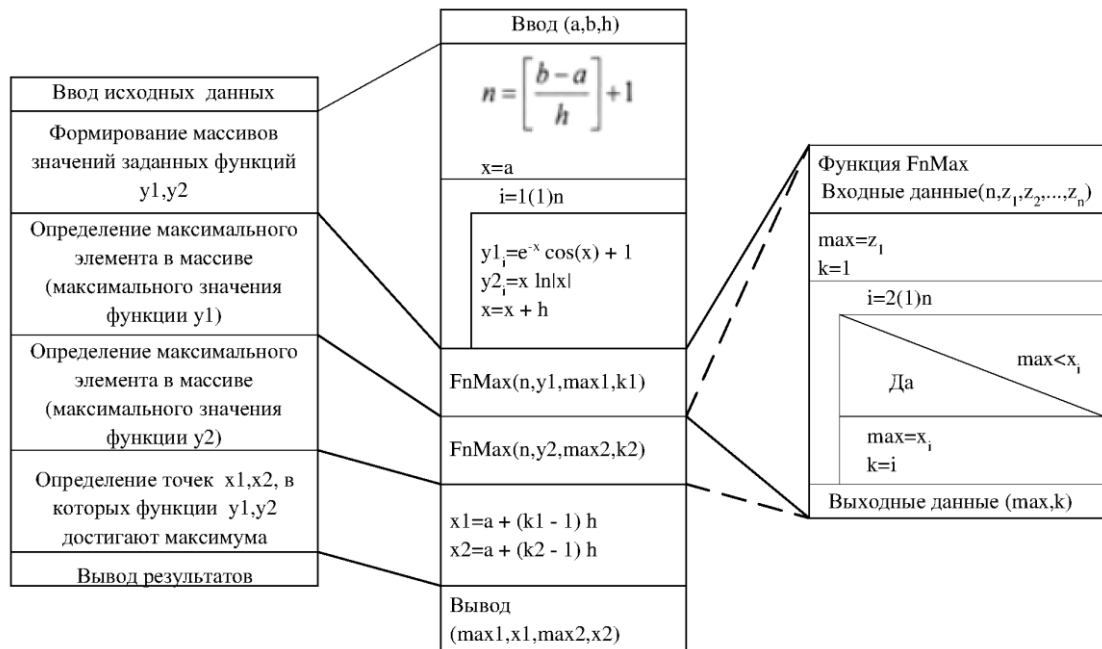


Рисунок 6. Структурограмма нисходящего проектирования алгоритма решения задачи 4

Программа имеет следующий данные:

Входные данные: a, b – границы отрезка, h – шаг изменения аргумента

Выходные данные: $max1, max2$ – максимальные значения функций; $x1, x2$ – точки, в которых функции имеют максимум.

Функция нахождения максимального элемента в одномерном массиве

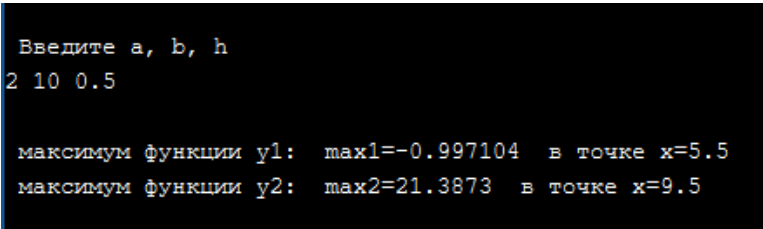
n – размер массива; z – имя массива;

max – максимальный элемент; k – номер максимального элемента.

Программа имеет следующий вид:

```
#include <iostream>
#include <math.h>
using namespace std;
#define Nmax 100
void FnMax(int, float [ ], float *, int *);
int main()
{
    int i, k1, k2;
    float a, b, h, max1, max2, x, x1, x2, y1[Nmax], y2[Nmax];
    cout<<"\n Введите a, b, h\n";
    cin>>a>>b>>h;
    int n=(int)(b-a)/h;
    /* Формирование массивов значений функций */
    for(x=a,i =0;x<=b;x+=h,i++)
    {
        y1[i] = exp(-x)*cos(x)-1;
        y2[i]= x*log(fabs(x));
    }
    FnMax(n, y1, &max1, &k1); /* Вызов функции */
    FnMax(n, y2, &max2, &k2); /* Вызов функции */
    x1=a+k1*h;
    x2=a+k2*h;
    cout<<"\n максимум функции y1: max1="<<max1<<" в точке x="<< x1;
    cout<<"\n максимум функции y2: max2="<<max2<<" в точке x="<< x2;
    return 0;
}

void FnMax(int n, float z[ ], float *max, int *k)
{
    int i;
    for(*max=z[0],*k =0,i =1;i<n;i++)
    if(*max < z[i])
    {
        *max = z[i];
        *k= i ;
    }
}
```



```
Введите a, b, h
2 10 0.5

максимум функции y1: max1=-0.997104 в точке x=5.5
максимум функции y2: max2=21.3873 в точке x=9.5
```

Рисунок 7. Результаты работы программы задачи 4

7. Закрепление нового материала, выполнение самостоятельной работы студентами

Задания выполняются студентами самостоятельно на ПК с использованием специализированного программного обеспечения – инструментальной среды программирования MS Visual Studio. Результаты выполнения каждого задания выборочно выводятся на интерактивную доску для проверки преподавателем (Приложение 5).

Методические указания и рекомендации к выполнению индивидуальной работы:

При модульном программировании сначала надо проанализировать общую задачу, определить вспомогательные задачи, записать их на обычный язык. Выбранная схема должна привести к решению задачи. Надо уделить внимание логической проверке связей между функциями, логике их работы при выполнении всего проекта задачи. Для каждой функции надо определить алгоритм вспомогательной задачи, области допустимых входных и выходных значений, а также возможные побочные эффекты (ошибки). Составить функциональную блок-схему решения задачи. Определить имя и тип значений каждой функции. Для каждой функции выписать все ее аргументы и полученные этой функцией результаты. Для каждого вызова выписать фактические значения аргументов и результатов, проанализировать механизм их передачи. Проанализировать схему обращений функций одна к другой. Найти аргументы всей задачи, обеспечить их ввод, позаботиться о результате всей задачи, обеспечить его вывод. Составляя программу, в начале привести прототипы всех функций в произвольном порядке.

Разработка многомодульных программ, состоящих из нескольких функций, чаще всего сопровождается ошибками, которые выявить иногда очень трудно. Эти ошибки чаще связаны с несоответствием между формальными и фактическими параметрами, их взаимодействием. При решении задач по данной теме надо быть очень внимательным в вопросах согласования формальных и фактических параметров.

Формальные и фактические параметры должны соответствовать по количеству и по типу. Проверка такой совместимости и преобразования при передаче значения фактического параметра в поле формального параметра переменных базовых типов осуществляется во время компиляции программы. Совместимость в последнем случае должна быть обеспечена самим программистом.

Студентам предлагаются разноуровневые задачи. Объясняется критерий оценивания результатов индивидуальной работы. Студент сам выбирает задание соответствующего уровня. С целью улучшения оценки студенты имеют возможность выполнить задание более сложного уровня дома.

8. Подведение итогов занятия

- анализ работы на занятии;
- выставление оценок; (Реестр работы студентов на занятии – приложение 7).
- рефлексия – осознание обучающимся и воспроизведение в речи того, что нового он узнал, чему научился на занятии, а что не удалось; заполнение рефлексивной мишени (приложение 6).

Студенты по кругу высказываются одним предложением, выбирая начало фразы из рефлексивного экрана на доске:

сегодня я узнал...	я приобрел...
было интересно...	я научился...
было трудно...	у меня получилось ...
я выполнял задания...	я смог...
я понял, что...	я попробую...
теперь я могу...	меня удивило...
я почувствовал, что...	мне захотелось...

10. Домашнее задание:

Задание 1. Составить программу вычисления $y = x^n$, используя функцию

$$\text{возведения вещественного числа в целую степень } z = a^m = \begin{cases} 1, m = 0 \\ a^m, m > 0 \\ \left(\frac{1}{a}\right)^{-m}, m < 0 \end{cases}$$

Задание 2. Составить программу вычисления наименьшего общего кратного двух введённых чисел x, y по формуле $NOK(a,b) = \frac{ab}{NOD(a,b)}$, используя функцию вычисления наибольшего общего делителя этих чисел.

Словарь терминов по теме «Функции в C++»

Глобальная переменная – переменная, определенная до объявления всех функций и доступная всем функциям.

Значение функции - результат вычислений функции.

Локальные переменные – переменные описанные внутри функции, а также переменные из списка аргументов. Область действия локальной переменной не выходит за рамки функции.

Модульное программирование — это организация программы как совокупности небольших независимых блоков, называемых модулями.

Модуль — функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом.

Описание функции – определение функции, включающее заголовок функции (тип возвращаемого значения, имя функции, типы и число формальных параметров) и тело функции

Параметр функции — это принятый функцией аргумент, значение, переданное из вызывающего кода.

Перегрузка функции – это возможность определять в одной программе несколько функций с одним и тем же именем, но с разными параметрами

Подпрограмма - самостоятельная часть программы, которая создаётся независимо от других частей и затем вызывается по имени. Когда имя подпрограммы используется в качестве оператора программы, выполняется вся группа операторов, представляющая тело подпрограммы.

Прототип функции - объявление функции, не содержащее тела функции, но указывающее имя функции, количество, типы аргументов и возвращаемый тип данных в качестве результата.

Рекурсивная функция — это функция, которая вызывает сама себя.

Сигнатура функции – это описание её заголовка, в которое обычно входят имя функции, число и тип и порядок следования передаваемых в неё параметров, тип возвращаемого значения

Тело функции – последовательность инструкций (операторов и описаний), предназначенных для решения некоторой задачи, заключенных в фигурные скобки

Фактический параметр — аргумент, передаваемый в функцию при ее вызове;

Формальный параметр — аргумент, указываемый при объявлении или определении функции.

Функция — это поименованный набор описаний и операторов, выполняющих определенную задачу.

Вопросы по теме «Функции в C++» для подготовки к занятию:

1. Дать определение функции. Для чего функции используются в программах?
2. Какую структуру имеет программа на C? Особенности функции `main()`.
3. Как объявить функцию?
4. Что такое описание и определение функции?
5. Какие виды функций имеются в C/C++? В чем их отличие?
6. Могут ли функции в C++ быть вложенными?
7. Как осуществляется выход из функции?
8. Для чего применяется оператор `return`?
9. Что такое прототип функции? Для чего используются прототипы функции? Всегда ли его присутствие необходимо в программе, содержащей функции?
10. Что называют формальными параметрами функции и что фактическими? Как эти два понятия соотносятся между собой?
11. Какие существуют способы передачи параметров в функцию?
12. При каком из способов передачи изменение формального параметра не влияет на изменение соответствующего ему фактического параметра?
13. Объясните, при каком способе передачи параметров изменение формального параметра влияет на изменение соответствующего ему фактического параметра?
14. Объясните, как необходимо действовать, если формальным параметром является указатель на переменную? Создайте функцию, которая обменивает значения двух переменных, передаваемых ей в качестве параметров. Что в этом случае будет использоваться в качестве фактических параметров?
15. Как правильно объявить и использовать аргументы по умолчанию?
16. Какие объекты являются локальными? Глобальными?
17. Почему не рекомендуется использовать глобальные объекты для передачи аргументов в функцию и результатов из неё?
18. Признаком чего служит слово `void`, стоящее перед именем функции?
19. Признаком чего служит слово `void`, стоящее на месте параметров? Всегда ли оно необходимо?
20. Как вызвать типизированную функцию? Нетипизированную функцию?
21. В чем разница между функциями, не возвращающими никаких значений и возвращающими значения объявленного типа?
22. В чем заключается особенность передачи массивов в качестве параметров функций?
23. Можно ли возвращать массив из функции, пользуясь оператором `return`?
24. Какие функции называются перегруженными? Для чего используется перегрузка функций?

Задания для самостоятельной подготовки

1. Составить программу вычисления данного выражения:
 $y = \frac{\min(x+y, y) + \min(x+z, y)}{1 - \min(x, z+0,2)}$, где $\min(a, b)$ - функция нахождения наименьшего из двух чисел.

2. Составить программу вычисления значения выражения:
 $\varphi = \sqrt{x^2 + y^2 + \sin^2 xy} + \sqrt{y^2 + z^2 + \sin^2 yz} + \sqrt{z^2 + x^2 + \sin^2 zx}$, используя функцию для вычисления значения $\sqrt{a^2 + b^2 + \sin^2 ab}$.

3. Напишите программу вычисления выражения:

$$y = \ln^2(1+x) + \cos(x+1) - e^x, \text{ где } x = \begin{cases} kz^3, k < 1 \\ z(z+1), k \geq 1 \end{cases}.$$

Вычисление значений x оформить в виде функции.

4. Составить программу вычисления данного выражения: $y = \frac{x^6(x-5)^3}{(2x+1)^5}$.

Возведение вещественного основания в степень с натуральным показателем оформить в виде функции, как нахождение произведения одинаковых множителей, не используя стандартной математической функции вычисления степени.

5. Составьте программу вычисления числа сочетаний из N по M . Число сочетаний определяется по формуле: $C_N^M = \frac{N!}{M!(N-M)!}$, где $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$.

Вычисление факториала оформить в виде функции.

6. Составить программу нахождения большего из 4 чисел a, b, c, d с использованием функции поиска большего из двух.

7. Даны действительные числа s и t . Получить $f = g(1.2, s) + g(t, s) - g(2s-1, st)$, где $g(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$. Вычисление величины $g(a, b)$ оформить в виде функции.

8. Даны действительные числа s и t . Получить $\varphi = f(t, -2s, 1.17) + f(2.2, t, s-t)$, где $f(a, b, c) = \frac{2a-b-\sin c}{5+|c|}$. Вычисление величины $f(a, b, c)$ оформить в виде функции.

9. Вычислить $Z = (\text{sign}(x) + \text{sign}(y)) \cdot \text{sign}(x+y)$, где $\text{sign}(a)$ - функция определения знака числа, принимающая значение 1, если число a - положительное; -1, если число a - отрицательное; 0 - при $a=0$.

10. Составить программу, которая вычисляет периметр треугольника, если заданы координаты вершин $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Вычисление расстояния между двумя точками оформить в виде функции по известной формуле:
 $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

11. Составить функцию вычисления высот треугольника со сторонами a, b, c по формулам $h_a = \frac{2S}{a}$, $h_b = \frac{2S}{b}$, $h_c = \frac{2S}{c}$, где $S = \sqrt{p(p-a)(p-b)(p-c)}$, $p = \frac{a+b+c}{2}$, если заданы стороны треугольника a, b, c .
12. Составить программу, которая проверяет, является ли четырёхугольник, заданный координатами своих вершин $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ ромбом. Вычисление длин сторон оформить в виде функции по известной формуле: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.
13. Вычислить $z = \frac{\max}{n_1} + \frac{\min}{n_2}$, где $\max$ и $\min$ – наибольшее и наименьшее значения элементов в массивах $A(N)$ и $B(M)$, состоящих из неповторяющихся значений, а n_1, n_2 – соответственно их порядковые номера в этих массивах.
14. Составить программу для подсчёта средних арифметических положительных элементов линейных таблиц $A[6]$ и $B[7]$. Использовать функции.
15. Составить программу для нахождения максимальных значений среди элементов линейных таблиц $A[13]$ и $B[8]$. Использовать функции.
16. Отрезок AB разбит на четыре равные части. Найти координаты точек деления, если $A(x_1, y_1), B(x_2, y_2)$. Вычисление координат середины отрезка оформить в виде функции.
17. Вычислить $z = \frac{s_1 + s_2}{k_1 k_2}$, где s_1 и k_1 – сумма и количество положительных элементов массива $X(N)$; s_2 и k_2 – сумма и количество положительных элементов массива $Y(M)$.
18. Вычислить $z = \frac{e^{s_1} + e^{s_2}}{k_1 k_2}$, где s_1 и k_1 – сумма и количество отрицательных элементов массива $A(N)$; s_2 и k_2 – сумма и количество отрицательных элементов массива $B(M)$.
19. Составить программу вычисления координат вектора $\vec{c} = 2\vec{a} + 3\vec{b}$, если заданы векторы $\vec{a}(a_1, a_2)$ и $\vec{b}(b_1, b_2)$. Вычисление координат вектора оформить в виде функции.
20. Составить программу нахождения суммы двух комплексных чисел $z_1 = x_1 + y_1 i$ и $z_2 = x_2 + y_2 i$ с использованием подпрограммы-процедуры вычисления $z = z_1 + z_2$.

Задачи повышенной сложности

1. Составить функцию определения косинуса угла между двумя векторами по формуле:

$\cos \varphi = \frac{\vec{a}\vec{b}}{|\vec{a}||\vec{b}|}$, где $\vec{a}\vec{b} = \sum_{i=1}^n a_i b_i$, $|\vec{x}| = \sqrt{\sum_{i=1}^n x_i^2}$. Для вычисления скалярного

произведения и модуля использовать функции. Найти косинусы углов между заданными векторами в трёхмерном пространстве. Исходные данные подобрать самостоятельно.

2. Составить функцию определения координат точки пересечения двух прямых $y = k_1 x + b_1$ и $y = k_2 x + b_2$, проходящих через заданные точки, по формулам: $x_0 = \frac{b_2 - b_1}{k_1 - k_2}$; $y_0 = \frac{k_1 b_2 - b_1 k_2}{k_1 - k_2}$. Коэффициенты k и b прямой $y = kx + b$,

проходящей через точки (x_1, y_1) , (x_2, y_2) , вычислить исходя из уравнения $\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$ с использованием функции. Исходные данные подобрать самостоятельно.

3. Составить функцию определения минимума функции $y = f(x)$, заданной в дискретных точках отрезка $[a, b]$ с постоянным шагом h . Используя её, найти минимумы функций: $y_1 = \sin^2 x \cos x - \ln|x|$, $y_2 = x^2 - 2x + 8$.

4. Составить функцию вычисления значения полинома n -го порядка по схеме Горнера. Используя её, вычислить значение функции: $z = \frac{12x^5 - 3x^3 + 2x^2 - 1}{6x^6 - 5x^4 + x - 8}$.

5. Составить программу вычисления определённого интеграла от функции $f(x)$ с использованием формул $\int_a^b f(x) dx \approx h \sum_{k=1}^n f(x_k)$; $h = (b - a) / n$; $x_k = a + kh$

(метод правых прямоугольников). Использовать функцию для вычисления значений $f(x)$ в точках разбиения отрезка интегрирования. Вычислить значение интеграла $\int_0^3 \sqrt{4 + x^2} dx$.

6. Разработать алгоритм и написать программу для нахождения решения системы линейных алгебраических уравнений $A\vec{x} = \vec{b}$ методом Крамера, где $A = (a_{ij})$ - квадратная матрица порядка $n=2$; $\vec{b} = (b_i)$ - вектор-столбец правой части уравнений системы. Оценить точность найденного решения. $\begin{cases} 4x_1 + 0,24x_2 = 8 \\ 0,09x_1 + 3x_2 = 9 \end{cases}$. Вычисление определителей оформить в виде функции.

7. Методом деления отрезка пополам найти приближённое значение корня уравнения $f(x) = 0$ на указанном отрезке. Абсолютная погрешность найденного значения не должна превосходить $\varepsilon = 10^{-3}$. Использовать функцию для вычисления значений $f(x)$ в точках.

$$x^5 - x - 0,2 = 0 , \quad [1; 1,1]$$

Компьютерное тестирование по теме «Функции C++»

Распределение баллов:

за правильные ответы на вопросы 1, 2, 3, 8, 10 – по 1 баллу

за правильные ответы на вопросы 4, 5, 6, 7, 9 – по 2 балла.

Максимальная сумма баллов 15 баллов.

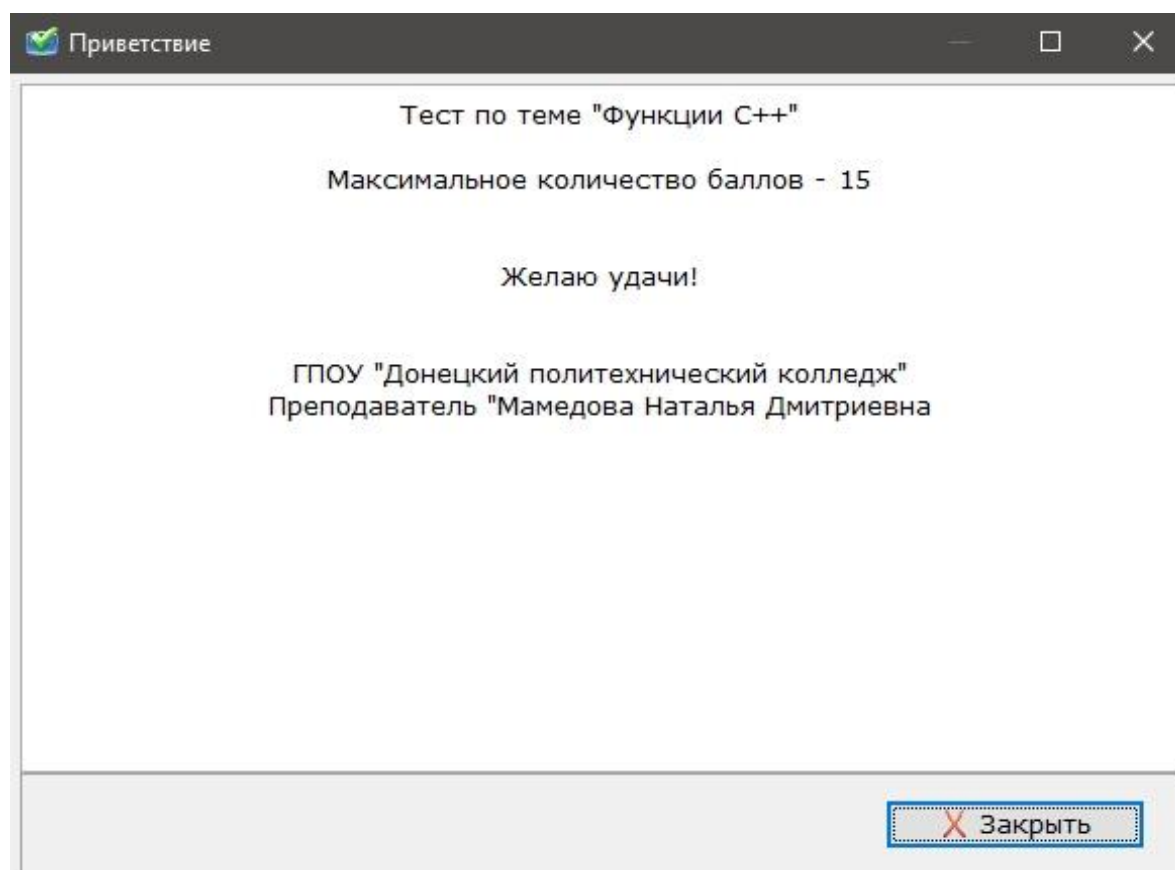
Критерии оценивания:

на «5» - 14-15 баллов

на «4» - 11-13 баллов

на «3» – 6-10 баллов

на «2» – ниже 6 баллов



Ответьте на вопрос

Вид Помощь

Тема № 3. Функции C++

Вопрос № 1

Если в функции на языке C++ отсутствует оператор return, то такая функция:

Выберите единственный правильный ответ

- ☐ 1. Будет возвращать значение 0
- ☐ 2. Не будет выполняться
- ☒ 3. Не будет возвращать значения
- ☐ 4. Будет возвращать значение 1

1 2 3 4 5 6 7 8 9 10

OK Пропустить Прервать

Всего вопросов: 10 Получено ответов: 0 Осталось вопросов: 10 0:00:03

Ответьте на вопрос

Вид Помощь

Тема № 3. Функции C++

Вопрос № 2

Установить последовательность в описании заголовка функции:

Установите последовательность ответов

3	Имя функции
2	Тип Функции
4	Список формальных параметров
1	Класс памяти

1 2 3 4 5 6 7 8 9 10

OK Пропустить Прервать

Всего вопросов: 10 Получено ответов: 1 Осталось вопросов: 9 0:00:13

Ответьте на вопрос

Вид

Помощь

Тема № 3. Функции C++

Вопрос № 3

Типизированная функция прекращает работу и возвращает значение с помощью оператора:

Выберите единственный правильный ответ

☐ 1. break;

☐ 2. return;

☒ 3. return <выражение>;

☐ 4. exit;

1

2

3

4

5

6

7

8

9

10

✓

OK

→

Пропустить

✗

Прервать

Всего вопросов: 10

Получено ответов: 2

Осталось вопросов: 8

0:00:33

Ответьте на вопрос

Вид

Помощь

Тема № 3. Функции C++

Вопрос № 4

Установить соответствие между понятием и определением этого понятия:

Установите соответствия ответов

...

1. Локальная переменная

...

2. Формальный параметр

...

3. Фактический параметр

...

4. Глобальная переменная

1

2

3

4

5

6

7

8

9

10

✓

OK

→

Пропустить

✗

Прервать

Всего вопросов: 10

Получено ответов: 3

Осталось вопросов: 7

0:01:29

34

Выбор соответствий

Ответ № 1. Локальная переменная

- ☒ А. переменная, определенная до объявления всех функций
- ☐ В. аргумент, указываемый при объявлении или определении функции
- ☐ С. аргумент, передаваемый в функцию при ее вызове
- ☒ D. переменная, описанная внутри функции

OK Отмена

Выбор соответствий

Ответ № 2. Формальный параметр

- ☒ А. аргумент, передаваемый в функцию при ее вызове
- ☐ В. переменная, определенная до объявления всех функций
- ☒ С. аргумент, указываемый при объявлении или определении функции
- ☐ D. переменная, описанная внутри функции

OK Отмена

Ответ № 3. Фактический параметр

- ☒ А. аргумент, передаваемый в функцию при ее вызове
- ☐ В. переменная, определенная до объявления всех функций
- ☐ С. аргумент, указываемый при объявлении или определении функции
- ☐ D. переменная, описанная внутри функции

✓ ОК

✗ Отмена

Ответ № 4. Глобальная переменная

- ☐ А. переменная, описанная внутри функции
- ☒ В. переменная, определенная до объявления всех функций
- ☐ С. аргумент, указываемый при объявлении или определении функции
- ☐ D. аргумент, передаваемый в функцию при ее вызове

✓ ОК

✗ Отмена

Ответьте на вопрос

Вид Помощь

Тема № 3. Функции C++

Вопрос № 5

Выбрать правильное(-ые) утверждение(-я):

Выберите возможные правильные ответы

- ☐ 1. В заголовке типизированной функции указывается тип возвращаемого значения void
- ☐ 2. В языке C++ подпрограммы делятся на процедуры и функции
- ☒ 3. Имя функции используется для вызова функции
- ☒ 4. Допускается создание функций, не имеющих аргументов, а также функций, не возвращающих никаких значений

1 2 3 4 5 6 7 8 9 10

OK Пропустить Прервать

Всего вопросов: 10 Получено ответов: 4 Осталось вопросов: 6 0:02:13

Ответьте на вопрос

Вид Помощь

Тема № 3. Функции C++

Вопрос № 6

Каким образом можно изменить фактический параметр в теле вызываемой функции?

Выберите возможные правильные ответы

- ☒ 1. Используя ссылки в качестве формального параметра
- ☐ 2. Использовать функцию без параметров
- ☒ 3. Используя указатели в качестве формального параметра
- ☐ 4. С помощью передачи параметра по значению

1 2 3 4 5 6 7 8 9 10

OK Пропустить Прервать

Всего вопросов: 10 Получено ответов: 5 Осталось вопросов: 5 0:02:27

Ответьте на вопрос

Вид Помощь

Тема № 3. Функции C++

Вопрос № 7

Выбрать неправильное(-ые) утверждение(-я):

Выберите возможные правильные ответы

☐ 1. Для использования библиотечных функций требуется включить с помощью директивы `#include`. в программу подзаголовочные файлы, содержащие их описания

☐ 2. Выполнение программы всегда начинается с функции с именем `main`

☒ 3. В программе с функциями обязательно использование прототипов функций

☒ 4. Если функция не использует параметров, то наличие круглых скобок не обязательно

1 2 3 4 5 6 7 8 9 10

☒ ☒ ☒ ☒ ☒ ☒ ☐ ☐ ☐ ☐

☒ ОК

Всего вопросов: 10 Получено ответов: 6 Осталось вопросов: 4 0:02:47

Ответьте на вопрос

Вид Помощь

Тема № 3. Функции C++

Вопрос № 8

Что такое прототип функции?

Выберите единственный правильный ответ

☐ 1. последовательность инструкций (операторов и описаний), предназначенных для решения некоторой задачи, заключенных в фигурные скобки

☐ 2. это функция, которая вызывает сама себя

☒ 3. объявление функции, не содержащее тела функции, но указывающее имя функции, количество, типы аргументов и возвращаемый тип данных в качестве результата

☐ 4. результат вычислений функции

1 2 3 4 5 6 7 8 9 10

☒ ☒ ☒ ☒ ☒ ☒ ☒ ☐ ☐ ☐

☒ ОК

Всего вопросов: 10 Получено ответов: 7 Осталось вопросов: 3 0:02:58

Ответьте на вопрос

Вид
 Помощь

Тема № 3. Функции C++

Вопрос № 9

Результатом какой функции является максимальное значение двух целочисленных переменных?

```

int max(int x, int y ) {
    if (x>y){
        return x;
    }
    return y;
}

int max(int x, int y ) {
    int m;
    if (x>y) m=x;
    else m=y;
    return m;
}

int max(int x, int y ) {
    int m;
    if (x<y) m=x;
    else m=y;
    return m;
}

int max(int x, int y ) {
    int m;
    if (x>y) m=x+y;
    else m=x-y;
    return m;
}

```

Выберите возможные правильные ответы

☒ 1. 1
☐ 2. 4
☒ 3. 2
☐ 4. 3

1

2

3

4

5

6

7

8

9

10

OK

Пропустить

Прервать

Всего вопросов: 10

Получено ответов: 8

Осталось вопросов: 2

0:04:42

Ответьте на вопрос

Вид
 Помощь

Тема № 3. Функции C++

Вопрос № 10

Двоечник решил переписать у одногруппника функцию вычисления факториала натурального числа N, но переписал с ошибками, переставив строки в неправильной последовательности. Запишите функцию в правильной последовательности:

Установите последовательность ответов

2	☐ int F = 1;
1	☐ int Fact(int N) {
4	☒ return F; }
3	☐ for(int i=1; i<=N; ++i) F = F* i;

1	2	3	4	5	6	7	8	9	10
☒	☒	☒	☒	☒	☒	☒	☒	☒	☒

ОК

Пропустить

Прервать

Всего вопросов: 10

Получено ответов: 9

Осталось вопросов: 1

0:04:59

Результаты

Тема № 3. Функции C++

Вопросы всего: **10**
 Задано вопросов: **10**
 Пропущено вопросов: **0**
 Правильных ответов: **10 (100%)**
 Неверных ответов: **0 (0%)**

Общее количество баллов: **10**
 Вы набрали баллов: **10 (100%)**

☒ Верно 100%
 ☐ Неверно 0%

Дата теста: 08.11.2020

Затраченное время: 0:05:04

БЕЗ ОШИБОК

Просмотр

Печать

Сохранить

Закрыть

Задания для практической работы студентов

Задание первого уровня

Составить программу решения поставленной задачи: составить таблицу значений функции $f(x)$ на отрезке $[a, b]$ с шагом h . Вычисление величины $f(x)$ с оформить в виде функции.

№ п/п	Функция $f(x)$	a	b	h
1	$x - \sin(x)$	0	$\pi/2$	$\pi/8$
2	$x + \cos(x)$	$\pi/4$	$\pi/2$	$\pi/16$
3	$x \cdot \cos(x)$	$\pi/3$	$2\pi/3$	$\pi/12$
4	$\operatorname{tg}(x) + x$	0	$\pi/4$	$\pi/16$
5	$x - \operatorname{ctg}(x)$	$\pi/4$	$\pi/2$	$\pi/16$
6	$\sin(x) + x$	0	1	0,1
7	$x - \operatorname{tg}(x)$	0.5	1	0,1
8	$x + \operatorname{ctg}(x)$	2	7	0,5
9	$\sin(x) - \cos(x)$	0	$\pi/2$	$\pi/8$
10	$x^2 \cdot \cos(x)$	$\pi/3$	$2\pi/3$	$\pi/6$
11	$\operatorname{tg}(x) + x^2$	0	$\pi/4$	$\pi/16$
12	$\operatorname{tg}(x) - x$	0	$\pi/4$	$\pi/16$
13	$x \cdot \sin(x)$	0	3π	$\pi/2$

Задание второго уровня

1. Используя функцию, для вычисления объёма цилиндра, найти в программе объёмы трех цилиндров с указанными параметрами и вывести наибольшее значение объёма.

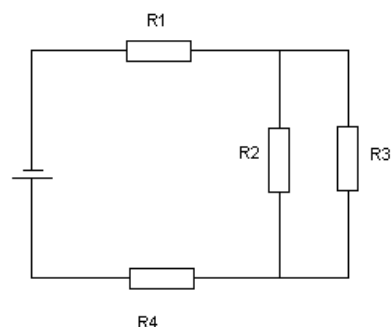
	<i>Радиус основания</i>	<i>Высота цилиндра</i>
<i>Первый цилиндр</i>	<i>R1</i>	<i>h</i>
<i>Второй цилиндр</i>	<i>a</i>	<i>10</i>
<i>Третий цилиндр</i>	<i>8.34</i>	<i>7.3</i>

Параметрами функции являются радиус основания и высота цилиндра.

```
// объём цилиндра
float veil(float h, float r)
{
    return(3.141593*r*r*h);
}
```

2. Используя функцию, которая вычисляет сопротивление электрической цепи, состоящей из двух резисторов, вычислить общее сопротивление четырех проводников с электрическими сопротивлениями, соединенных между собой по схеме, показанной на рисунке:

Параметрами функции являются величины сопротивлений и тип соединения (последовательное или параллельное). Функция проверяет корректность параметров: если неверно указан тип соединения, то функция возвращает -1.



// вычисляет сопротивление электрической цепи

```
float sopr( float r1, float r2, int t)
(
    // r1,r2 – величины сопротивлений, t – тип соединения:
    // 1 - последовательное; 2 - параллельное.
    float r;
    if (t==1) r=r1+r2;
    else if (t==2) r=r1*r2/(r1+r2);
    else r=-1;
    return (r);
}
```

3. Используя функцию, которая сравнивает два целых числа и возвращает результат сравнения в виде знака: >, < или =, вычислить количество равных среди введенных трёх чисел x, y, t.

// Функция compare возвращает результат сравнения двух чисел

// в виде символа отношения

```
char compare(int a, int b)
{
    char res;
```

```

if (a > b) res = '>';
else if (a < b) res = '<';
else res = '=';
return(res);
}

```

4. Банк предлагает договор сроком на 12 месяцев, в соответствии с которым физическим лицам насчитывается процентная ставка по окончании срока договора в зависимости от первоначальной суммы вклада: от 2000 до 3000 руб. - 4%, от 3000 до 10000 руб. – 6,5%, свыше 10000 руб. - 8%. Определить сумму вклада после завершения договора, если известна первоначальная сумма вклада. При решении задачи использовать функцию Procent, которая находит указанное число процентов от полученного в качестве аргумента числа

```

// Функция вычисления процента от числа x
double Procent (double x, double pr)
{
return x*pr/100;
}

```

5. Использовать функцию вычисления факториала factor и составить программу, которая обращается к функции для вывода таблицы факториалов для чисел из отрезка [m, n].

```

// вычисление факториала
unsigned int factor(int x)
{
unsigned int f = 1;
for (int i = 2; i <= x; i++)
f *= i;
return(f);
}

```

6. Используя функцию Dohod, которая вычисляет доход по вкладу, вычислить прибыль для следующих исходных данных:

	Сумма вклада	Процентная ставка	Срок вклада
Первый взнос	1000.00 руб.	3%	365 дней
Второй взнос	300 у.е.	5%	2 года
Третий взнос	5,5 тис. руб.	4,5%	100 дней

Исходными данными для функции являются величина вноса, процентная ставка (годовых) и срок вклада (количество дней).

```
// функция вычисляет доход по вкладу
float dohod(float sum, // сумма вклада
float stavka, // процентная ставка (годовых)
int srok) // срок вклада (дней)
{
return(sum*(stavka/100/365)*srok); // 365 количество дней в году
}
```

7. Для трёх данных уравнений $2x^2 + 5x + 2 = 0$, $2x^2 + 3x + 2 = 0$, $x^2 + 3x + 2 = 0$ вывести количество действительных корней. Использовать функцию, которая по известным коэффициентам квадратного уравнения находит количество его действительных корней (2-два различных корня, 1 - одинаковые корни, 0 - уравнение не имеет действительных корней). Если исходные данные неверны, то функция возвращает -1.

```
int kvadur(float a, float b, float c) // коэффициенты уравнения
{
float d; // дискриминант
if (a == 0) return(-1);
d = b*b-4*a*c;
if (d < 0) return(0);
else if (d==0) return (1);
else return (2);
}
```

8. Используя функцию NOD нахождения наибольшего общего делителя двух натуральных чисел, сократить данную обыкновенную дробь $\frac{m}{n}$.

```
// возвращает наибольший общий делитель двух чисел
int NOD(int v1, int v2)
{
while (v2)
{
int temp=v2;
v2=v1%v2;
v1=temp;
}
return v1;
}
```

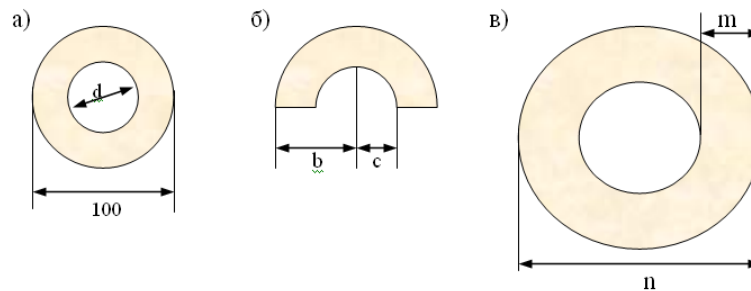
9. Используя функцию uear проверки является ли год високосным, вычислить, сколько членов вашей семьи родились в високосные годы. Параметром функции является номер года, результат логического типа.

```

// функция определяет, является ли год високосным
bool year(unsigned int x)
{
    if (x%4) return false;
    else return true;
}

```

10. Используя функцию, которая вычисляет площадь кольца по известным внешнему и внутреннему радиусам, вычислить площади трёх фигур, которые изображены на рисунках:



```

// функция вычисления площади кольца
float plosh(float r1, float r2) // радиусы кольца
{
    float s; // площадь кольца
    if (r1>r2)
        s=3.14*(r1*r1-r2*r2);
    else s=3.14*(r2*r2-r1*r1);
    return s;
}

```

11. Составить программу, которая определяет, можно ли построить треугольник со следующими сторонами а) 3, 7, 9; б) x , y , z ; в) $2x$, $3y$, $x + y$. Для проверки, существует треугольник с данными сторонами, использовать функцию `treugolnik`. В качестве аргументов функции выступают стороны, результат логического типа.

// функция определяет, существует ли треугольник с данными сторонами

```

bool treugolnik(float a, float b, float c)
{
    bool flag=false;
    if (a<b+c && b<a+c && c<a+b) flag=true;
    return flag;
}

```

12. Составить программу вычисления количества простых чисел, принадлежащих отрезку $[m, n]$. Для проверки, является ли число простым,

использовать функцию `prostoe`. Функция возвращает 1, если число является простым, а 0 - в противном случае.

// функция проверяет, является ли число простым

```
int prostoe(int n)
{
    int d, r;
    d=2;
    do {
        r=n%d;
        if (r!= 0) d++;
    }
    while (r!=0); // пока n не поделится на d
    if (d==n) return 1;
    else return 0;
}
```

13. Используя функцию, которая вычисляет площадь поверхности цилиндра, вычислить площади поверхностей трех цилиндров по данным параметрам и найти среди них наименьшую площадь. Параметрами функции являются радиус и высота цилиндра.

	<i>Радиус основания</i>	<i>Высота цилиндра</i>
<i>Первый цилиндр</i>	<i>R</i>	<i>H</i>
<i>Второй цилиндр</i>	3.56	7
<i>Третий цилиндр</i>	<i>c</i>	<i>v</i>

// вычисление площади поверхности цилиндра

```
float cilindr(float r, float h)
{
    float s; // площадь поверхности цилиндра
    s = 2*3.14*r*r-2*3.14*r*h;
    return s;
}
```

Задание третьего уровня

1. Составить программу, которая проверяет, есть ли среди векторов, заданных своими координатами, перпендикулярные. Условие перпендикулярности двух векторов $\vec{a}(a_1, a_2)$ и $\vec{b}(b_1, b_2)$ - скалярное произведение таких векторов равняется 0, т.е. $\vec{a} \cdot \vec{b} = 0$. Вычисление

скалярного произведения векторов оформить при помощи функции по формуле $\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2$.

2. Составить программу для вычисления медиан треугольника с заданными

$$m_a = 0,5\sqrt{2b^2 + 2c^2 - a^2}$$

сторонами a, b, c по формулам: $m_b = 0,5\sqrt{2a^2 + 2c^2 - b^2}$.

$$m_c = 0,5\sqrt{2a^2 + 2b^2 - c^2}$$

3. Составить программу, которая определяет углы треугольника со

$$\cos \alpha = \frac{b^2 + c^2 - a^2}{2bc}$$

сторонами a, b, c по известным формулам: $\cos \beta = \frac{a^2 + c^2 - b^2}{2ac}$.

$$\cos \gamma = \frac{a^2 + b^2 - c^2}{2ab}$$

4. Составить программу, которая находит наибольшую сторону треугольника, если заданы координаты вершин $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Вычисление расстояния между двумя точками оформить в виде функции по известной формуле: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

5. Составить программу, которая проверяет, есть ли среди векторов $\vec{v}, \vec{s}, \vec{t}$, заданных своими координатами, коллинеарные. Условие коллинеарности двух векторов $\vec{a}(a_1, a_2)$ и $\vec{b}(b_1, b_2)$ - соответствующие координаты векторов прямо пропорциональны, т.е. $\frac{a_1}{b_1} = \frac{a_2}{b_2}$. Проверку этого условия оформить с помощью функции.

6. Составить программу вычисления данного выражения: $y = \frac{|x^2 - 6x| + 2|x|}{|x^3 + 3x^2 - 5x| + 4}$

Вычисление значения модуля оформить в виде функции $|x| = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$.

7. Составить функцию, которая вычисляет радиус r окружности, вписанной в правильный многоугольник с известной длиной стороны a и известным числом сторон n по формуле $r = \frac{a}{2 \operatorname{tg} \frac{180^\circ}{n}}$. Найти величину радиуса

вписанной окружности r для $n = 3, 4, 6$.

8. Составить программу для вычисления $Z = \frac{V_1 + V_2 + V_3}{3}$, где V_1, V_2, V_3 - объёмы шаров с радиусами r_1, r_2, r_3 . Вычисление объёма шара по формуле $V = \frac{4}{3}\pi R^3$ оформить в виде функции.

9. Составить программу вычисления данного выражения:
 $y = \frac{\max(a, b) + \max(b, c)}{\max(a, c)}$, где $\max(x, y)$ - функция нахождения наибольшего из двух чисел.

10. Составить программу, которая вычисляет биссектрисы треугольника со сторонами a, b, c по формулам:

$$l_a = \frac{\sqrt{bc(a+b+c)(b+c-a)}}{b+c}$$

$$l_b = \frac{\sqrt{ac(a+b+c)(a+c-b)}}{a+c}$$

$$l_c = \frac{\sqrt{ab(a+b+c)(a+b-c)}}{a+b}$$

11. Составить программу, которая для четырёхугольника, заданного координатами своих вершин $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$, проверяет, является ли он ромбом. Вычисление длин отрезков оформить в виде функции по известной формуле: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

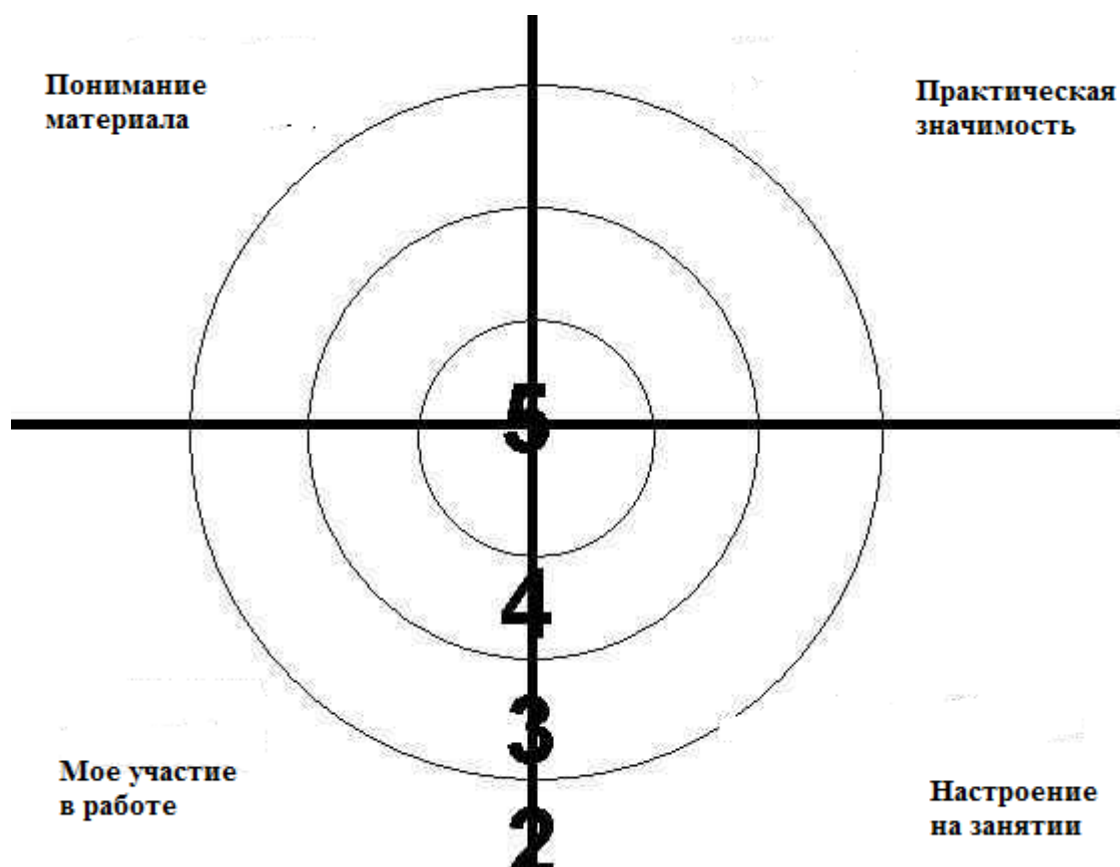
12. Составить функцию, которая вычисляет радиус R окружности, описанной около правильного многоугольника с известной длиной стороны a и известным числом сторон n по формуле $R = \frac{a}{2 \sin \frac{180^\circ}{n}}$. Найти величину

радиуса описанной окружности R для $n = 3, 4, 6$.

13. Составить программу с использованием функции, которая проверяет, сколько существует треугольников с предлагаемыми длинами сторон a, b, c, d . Для проверки существования треугольника со сторонами a, b, c

использовать неравенство треугольника:
$$\begin{cases} a < b + c \\ b < a + c \\ c < a + b \end{cases}$$

Рефлексивная мишень(заполняется каждым студентом)



ПРИЛОЖЕНИЕ 7

Реестр работы студентов за занятия

№ п\п	Фамилия, имя студента	Вид проверки знаний			
		Домашнее задание	Компьютерное тестирование	Индивидуальное задание	Оценка
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					

Дата проведения
занятия: