

**Компонент ОПОП 01.03.02 Прикладная математика и информатика
Направленность (профиль) Системное программирование и компьютерные технологии**

Б1.О.15.04

РАБОЧАЯ ПРОГРАММА

**Дисциплины
(модуля)**

Языки и технологии программирования

Разработчики:
Козинец Егор Анатольевич,
доцент кафедры информационных
технологий;

Королева Наталья Юрьевна,
доцент кафедры
информационных технологий,
канд. пед. наук, доцент

Утверждено на заседании кафедры
Информационных технологий
протокол № 6 от 01.02.2024

Заведующий кафедрой ИТ

_____ О.И. Ляш

**Мурманск
2024**

Пояснительная записка

Объем дисциплины 3 з.е.

1. Результаты обучения по дисциплине (модулю), соотнесенные с индикаторами достижения компетенций, установленными образовательной программой

Компетенции	Индикаторы достижения компетенций	Результаты обучения по дисциплине (модулю)
<p>ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач</p> <p>ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</p>	<p>ИД-1ОПК-2 Использует и адаптирует существующие математические методы для разработки и реализации алгоритмов решения прикладных задач</p> <p>ИД-2ОПК-2 Использует существующие системы программирования для разработки и реализации алгоритмов решения прикладных задач.</p> <p>ИД-1ОПК-5 Разрабатывает алгоритмы решения практических задач в области профессиональной деятельности.</p> <p>ИД-2ОПК-5 Способен разрабатывать компьютерные программы пригодные для практического применения.</p> <p>ИД-3ОПК-5 Применяет алгоритмы и компьютерные программы для решения практических задач в области профессиональной деятельности.</p>	<p><i>Знать:</i></p> <ul style="list-style-type: none"> – парадигмы программирования и реализующие их конкретные технологии; – основные характеристики языка программирования, определения алфавита, синтаксиса и семантики; основные этапы эволюции языков и технологий программирования; понятие и состав среды разработки; – способы человеко-машинного взаимодействия при решении задач на ЭВМ; этапы решения задач на ЭВМ и жизненного цикла программного продукта; – виды модулей в составе проекта (совокупности файлов, обеспечивающих решение задачи на ЭВМ); – процессы, происходящие при сборке проекта, и реализующие их утилиты; – понятия отладки, тестирования, верификации и валидации программ, виды ошибок; критерии оценки качества программы; понятие спецификации программы; – определение алгоритма и его свойства, способы записи алгоритма, виды структур алгоритмов; – назначение и правила оформления основных алгоритмических конструкций процедурного программирования; – основные типы данных, их назначение, ограничения, допустимые операции; способы определения подпрограмм и передачи им параметров, ограничения на типы возвращаемых значений; – способы организации массивов; – приведение и вывод базовых типов данных; – модель работы с выделяемой для программы памятью: статический раздел, стек, куча; – динамическое выделение и освобождение памяти; – механизмы ссылок и указателей; – агрегатные (контейнерные) пользовательские типы данных (перечисления, структуры, массивы, строки, различные виды списков, словари, хэш-таблицы); – стандартные алгоритмы работы с агрегатными типами; – операции для работы с файлами входных и выходных данных (текстовыми и бинарными); – модули стандартных библиотек для выполнения операций структурного программирования; примеры сторонних (инструментальных) библиотек и задач, которые могут быть решены с их помощью средствами языка программирования. <p><i>Уметь:</i></p> <ul style="list-style-type: none"> – анализировать условие задачи на целесообразность применения той или иной технологии программирования; – выделять отдельные подзадачи в соответствии с

		<p>выбранной технологией;</p> <ul style="list-style-type: none"> – определять, какие действия, на каком этапе разработки программы выполняются; – выделять при решении задачи наиболее важные критерии качества программы; – осуществлять отладку программ различными способами; – выделять и обрабатывать исключительные ситуации (ошибки) вычислительного процесса; – составлять план тестирования, формировать и документировать тестовые наборы; – представлять алгоритмы различными способами; – оценивать вычислительную (временную) и объёмную (пространственную) сложность алгоритмов; – осуществлять программную реализацию алгоритмов на выбранном языке программирования в среде разработки; – выполнять рефакторинг, оптимизацию производительности, реинжиниринг уже существующей программы; – осуществлять работу с базовыми и пользовательскими типами данных; – реализовывать алгоритмы чтения, обработки, записи данных во внешний файл; – создавать многомодульные проекты, подключать модули стандартной библиотеки, подключать сторонние библиотеки. <p><i>Владеть:</i></p> <ul style="list-style-type: none"> – терминологией для определения и описания этапов жизненного цикла программ; – навыками выделения и формулирования этапов решения задач на ЭВМ в контексте выбранной технологии программирования; – технологией реализации процедурной (структурной) парадигмы программирования средствами выбранной технологии программирования; – навыками сборки, отладки, тестирования и документирования программ; – навыками работы с отдельными типами данных.
--	--	---

2. Содержание дисциплины (модуля)

Раздел 1. Парадигмы, языки и технологии программирования

1.1. Основные понятия языков программирования и сред разработки.

Язык программирования: семантика, синтаксис, алфавит, идентификатор, объект, представление, переменная, константа, литерал. Жизненный цикл программы. Компоненты среды разработки.

1.2. Эволюция парадигм, языков и технологий программирования.

Императивное программирование: структурное, процедурное, объектно-ориентированное, агентно-ориентированное, обобщённое. Декларативное программирование: функциональное, логическое. Прикладные математические пакеты (среды математического программирования). Краткая история развития языков программирования. Среды исполнения JRE и .NET CLR.

Раздел 2. Разработка программ и реализация основных алгоритмических конструкций средствами выбранного языка программирования и среды разработки.

2.1. Сборка проектов на языке C++ в среде Qt Creator. Модули в составе проекта: исходного текста, заголовочные, объектные, исполняемые. Утилиты сборки: make, qmake,

препроцессор компилятора, утилиты для компиляции файлов графического интерфейса, классов, ресурсов. Директивы препроцессора. Работа с отладчиком.

2.2. Введение в синтаксис C++. Стандартная библиотека языка C++ и её заголовочные файлы. Операторы ввода/вывода. Функция main. Встраиваемые функции. Макроопределения для функций. Идентификаторы для переменных и функций: рекомендации по названиям, объявления, определения, инициализация. Синтаксис C как основа синтаксиса языков C++, Java, C# (сходства и различия).

2.3. Базовые (фундаментальные) типы данных. Логический, символьный, целочисленные, вещественные типы: множества значений и допустимые операции. Типы с фиксированным размером. Примеры литералов для различных типов. Указатели и ссылки. Константность.

2.4. Инструкции и операторы. Инструкции выбора (ветвления) и цикла. Инструкции break и continue. Понятие оператора. Приоритет операторов.

2.5. Функции. Объявление и определение функции. Способы передачи аргументов в функцию. Объявление статических переменных в теле функции. Рекурсия.

Раздел 3. Обработка отдельных типов данных средствами выбранного языка программирования и среды разработки.

3.1. Приведение типов. Использование указателей и ссылок.

Явное и неявное приведение базовых типов. Вывод типов. Пользовательские типы данных: структуры, перечисления. Указатели и ссылки. Указатели при работе с массивами. Тип данных вектор.

3.2. Работа с файлами и строками.

Инструкции чтения и записи данных для текстовых и бинарных файлов. Действия со строками. Обработка массивов строк.

3.3. Библиотека STL и сторонние инструментальные библиотеки. Агрегатные типы данных (контейнеры). Шаблоны. Стандартные алгоритмы обработки данных для типов из библиотеки STL. Создание многомодульных проектов. Подключение и использование API сторонних библиотек (на примере OpenGL и т.д.).

3. Перечень учебно-методического обеспечения дисциплины (модуля)

- мультимедийные презентационные материалы по дисциплине (модулю) представлены в электронном курсе в ЭИОС МАУ;
- методические указания к выполнению лабораторных/практических/контрольных работ (выбрать) представлены в электронном курсе в ЭИОС МАУ;
- методические материалы для обучающихся по освоению дисциплины (модуля) представлены на официальном сайте МАУ в разделе «Информация по образовательным программам, в том числе адаптированным».

4. Фонд оценочных средств по дисциплине (модулю)

Является отдельным компонентом образовательной программы, разработан в форме отдельного документа, включает в себя:

- перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины (модуля);
- задания текущего контроля;
- задания промежуточной аттестации;
- задания внутренней оценки качества образования.

5. Перечень основной и дополнительной учебной литературы (печатные издания, электронные учебные издания и (или) ресурсы электронно-библиотечных систем).

Основная литература

1. Белоцерковская, И.Е. Алгоритмизация. Введение в язык программирования С++: учеб. пособие / И.Е. Белоцерковская, Н.В. Галина, Л.Ю. Катаева. – 2-е изд., испр. – М. : Национальный Открытый Университет «ИНТУИТ», 2016. – 197 с.; То же [Электронный ресурс]. – URL: <http://biblioclub.ru/index.php?page=book&id=428935>
2. Головин, И.Г. Языки и методы программирования : учебник для студ. вузов, обуч. по направл. «Прикладная математика и информатика» / И.Г. Головин, И.А. Волкова. – М. : Академия, 2012.–303 с.
3. Кирнос, В.Н. Информатика II. Основы алгоритмизации и программирования на языке С++ : учебно-методическое пособие / В.Н. Кирнос ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). – Томск : Эль Контент, 2013. – 160 с.; То же [Электронный ресурс]. – URL: <http://biblioclub.ru/index.php?page=book&id=208651>.

Дополнительная литература:

4. Программирование на языке С++ в среде Qt Creator: учебно-методическое пособие / Г.Г. Злобин, Д.А. Костюк, А.С. Чмыхало и др. – 2-е изд., испр. – М. : Национальный Открытый Университет «ИНТУИТ», 2016. – 716 с.; То же [Электронный ресурс]. – URL: <http://biblioclub.ru/index.php?page=book&id=428929>.
5. Орлов, С.А. Теория и практика языков программирования : [для бакалавров и магистров] : учебник по направл. «Информатика и вычислительная техника» / С.А. Орлов – СПб. : Питер, 2013. - 688 с.
6. Страуструп, Б. Программирование: принципы и практика с использованием С++ / Б. Страуструп; Пер. с англ. – 2-е изд. – М. : Вильямс, 2016. – 1328 с.

6. Профессиональные базы данных и информационные справочные системы

- 1) *Государственная система правовой информации - официальный интернет-портал правовой информации- URL: <http://pravo.gov.ru>*
- 2) *Информационная система «Единое окно доступа к образовательным ресурсам» - URL: <http://window.edu.ru>*
- 3) *Справочно-правовая система. Консультант Плюс - URL: <http://www.consultant.ru/>*

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства

- 1) Офисный пакет Microsoft Office 2007
- 2) Система оптического распознавания текста ABBYY FineReader
- 3) Adobe Reader
- 4) FlashPlayer
- 5) Git
- 6) Google Chrome
- 7) K-Lite Codec Pack
- 8) LibreOffice.org
- 9) Mozilla FireFox
- 10) Google Chrome
- 11) NetBeans
- 12) Notepad++
- 13) Paint.NET
- 14) Python 3.6 Anaconda
- 15) StarUML
- 16) The Gimp
- 17) VirtualBox
- 18) VLC

8. Обеспечение освоения дисциплины лиц с инвалидностью и ОВЗ

Обучающиеся из числа инвалидов и лиц с ОВЗ обеспечиваются печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

9. Материально-техническое обеспечение дисциплины (модуля) представлено в приложении к ОПОП «Материально-технические условия реализации образовательной программы» и включает:

- учебные аудитории для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенные оборудованием и техническими средствами обучения;

- помещения для самостоятельной работы обучающихся, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде МАУ;

Допускается замена оборудования его виртуальными аналогами.

10. Распределение трудоемкости по видам учебной деятельности

Таблица 1 - Распределение трудоемкости

Вид учебной учебной деятельности	Распределение трудоемкости дисциплины (модуля) по формам обучения											
	Очная			Очно-заочная				Заочная				
	Семестр			Всего часов	Семестр			Всего часов	Семестр/Курс			Всего часов
	2											
Лекции	14										14	
Лабораторные работы	30										30	
Самостоятельная работа	64										64	
Всего часов по дисциплине	108										108	
/ из них в форме практической подготовки												

Формы промежуточной аттестации и текущего контроля

зачет	За										
-------	----	--	--	--	--	--	--	--	--	--	--

Перечень лабораторных работ по формам обучения

№ п/п	Темы лабораторных работ
1	2
	Очная форма
1	Создание первого проекта на языке C++ в среде Qt. Решение СЛАУ методом Крамера.
2	Использование основных синтаксических конструкций языка C++: ввод/вывод и

	управляющие инструкции
3	Использование основных синтаксических конструкций языка C++. Решение СЛАУ размером (3×3) методами Крамера и Гаусса-Жордана. Вычисление определителей малых порядков
4	Средства отладки Qt. Способы передачи параметров в функцию. Работа с текстовыми файлами. Вычисление значения детерминанта конечного порядка по определению
5	Способы реализации массивов и простейших списков
6	Обработка строк
7	Создание оконного приложения
8	Подключение сторонних библиотек (на примере OpenGL).