

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Физический факультет

Утверждено

решением Ученого совета ННГУ
протокол от «31» мая 2023 г. № 6

Рабочая программа дисциплины

Информатика

Уровень высшего образования
бакалавриат

Направление подготовки / специальность
09.03.02 Информационные системы и технологии

Направленность образовательной программы
**Информационные системы и технологии в физических
исследованиях**

Форма обучения
очная

Год начала подготовки

2022 год

Нижний Новгород

2023год

1. Место дисциплины в структуре ООП

Дисциплина «Информатика» (Б1.О.06) относится к обязательной части Блока 1 «Дисциплины (модули)» учебного плана ООП.

Дисциплина преподается в 1 и 2 семестре.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции* (код, содержание индикатора)	Результаты обучения по дисциплине**	
ОПК-1. Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1. Знать основы высшей математики, общей физики, теории вероятности и технологий программирования.	<i>Знать</i> общие характеристики процессов сбора, передачи, обработки и накопления информации, правила использования операторов, объявлений и определений языка программирования высокого уровня C++, правила вычисления выражений, в том числе правила приоритета операций языка программирования C++.	Тест, задача (практическое задание), зачет, экзамен
	ОПК-1.2. Уметь решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования.	<i>Уметь</i> логически верно описывать действия программы на алгоритмическом языке, создавать приложения для ОС Windows с использованием языка программирования C++, грамотно использовать операторы языка C++.	Задача (практическое задание), зачет, экзамен
	ОПК-1.3. Иметь навыки теоретического и экспериментального исследования объектов профессиональной деятельности.	<i>Владеть</i> средствами описания объектов и процессов реального мира языком программирования высокого уровня, инструментами разработки приложений, входящими в пакет Microsoft Visual Studio.	Задача (практическое задание)
ОПК-3. Способен решать стандартные задачи профессиональной	ОПК-3.1. Знать методы решения стандартных задач профессиональной	<i>Знать</i> основные конструкции структурного и объектно-ориентированного программирования языка C++, основные этапы создания	Тест, задача (практическое задание), зачет, экзамен

<p>деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</p>	<p>деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p>	<p>программы на алгоритмическом языке (проектирование, кодирование, тестирование, отладка), особенности модели программирования для ОС Windows, основные классы библиотеки Microsoft Foundation Class Library (MFC), принципы организации вывода текста и графики в приложении Windows.</p>	
	<p>ОПК-3.2. Уметь решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p>	<p><i>Уметь</i> создавать алгоритмы для численного моделирования разнообразных объектов и систем, создавать собственные типы данных и наборы функций для взаимодействия с объектами этих типов, выбирать правильную архитектуру построения приложения MFC, использовать программный интерфейс графического устройства ОС Windows (GDI).</p>	<p>Задача (практическое задание), зачет, экзамен</p>
	<p>ОПК-3.3. Иметь навыки подготовки обзоров, аннотаций, составления рефератов, научных докладов.</p>	<p><i>Владеть</i> мастерами автоматической генерации кода MFC, эффективными способами кодирования и документирования программы, методами решения функциональных и вычислительных задач.</p>	<p>Задача (практическое задание)</p>
<p>ПК-2. Способен проводить научные исследования и управлять результатами научно-исследовательских работ при разработке и внедрении информационных технологий и систем</p>	<p>ПК-2.1. Знать особенности проведения научных исследований.</p>	<p><i>Знать</i> основные этапы создания программы на алгоритмическом языке (проектирование, кодирование, тестирование, отладка).</p>	<p>Тест, задача (практическое задание), зачет, экзамен</p>
	<p>ПК-2.2. Уметь обрабатывать и анализировать результаты научно-исследовательских работ.</p>	<p><i>Уметь</i> создавать алгоритмы для численного моделирования разнообразных объектов и систем, создавать собственные типы данных и наборы функций для взаимодействия с объектами этих типов.</p>	<p>Задача (практическое задание), зачет, экзамен</p>
	<p>ПК-2.3. Владеть навыками разработки и внедрения информационных технологий в системах научных исследований.</p>	<p><i>Владеть</i> средствами описания объектов и процессов реального мира языком программирования высокого уровня.</p>	<p>Задача (практическое задание)</p>

3. Структура и содержание дисциплины

3.1. Трудоемкость дисциплины

Очная форма обучения	
Общая трудоемкость	7 ЗЕТ
Часов по учебному плану	252
в том числе	
аудиторные занятия (контактная работа):	115
- занятия лекционного типа, ч	48
- практические занятия, ч	64
- лабораторных, ч	
- КСРИФ, ч	3
контроль	36
самостоятельная работа, ч	101
Промежуточная аттестация	зачет в 1 семестре и экзамен во 2 семестре

3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины (модуля),	Всего (часы)	В том числе				Самостоятельная работа, часы
		Контактная работа (работа во взаимодействии с преподавателем), часы				
		из них				
		Занятия лекционного типа	Практические занятия	Занятия лабораторного типа	Всего	
1. Базовые средства языка С++. Алфавит языка. Основные типы данных С++. Структура программы на С++. Базовые конструкции структурного программирования. Типы данных, определяемые пользователем.	38	8	8		16	22
2. Модульное программирование. Функции. Способы взаимодействия функций.	26	8	8		16	10

Перегрузка и шаблоны функций. Директивы препроцессора. Пространства имен.						
3. Объектно-ориентированное программирование. Основные свойства ООП. Описание классов и объектов. Перегрузка операций. Наследование. Шаблоны классов. Обработка исключений.	38	8	8		16	22
4. Стандартная библиотека C++. Потокные классы. Стандартные и файловые потоки. Строковый класс. Последовательные и ассоциативные контейнеры. Итераторы.	26	8	8		16	9
5. Введение в программирование для ОС Windows. История возникновения и развития ОС Windows. Модель программирования Windows. Типы данных Windows. Классы библиотеки MFC.	24	3	6		9	7
6. Обработка сообщений в MFC. Обработка сообщений в ОС Windows. Группы сообщений. Карта сообщений MFC.	24	3	6		9	7
7. Диалоговые панели. Типы диалоговых панелей. Редактирование и отображение диалоговой панели. Динамический обмен данными с диалоговой панелью.	24	3	6		9	7
8. Вывод текста и графики в MFC. Интерфейс графического устройства. Классы контекста устройства. Время жизни и состояние контекста устройства. Классы графических объектов.	24	3	6		9	7
9. Архитектура «Документ – Представление». Понятие	28	4	8		12	10

документа и представления. Шаблон документа. Отношения между объектами архитектуры. Сериализация.						
<u>Итого</u>	252	Error! Re source not	Error! Re source not	Error! Re source found.	Error! Re source not f	101

Текущий контроль успеваемости реализуется в рамках занятий семинарского типа, групповых или индивидуальных консультаций.

Промежуточная аттестация проходит в традиционной форме -зачет в 1 семестре и экзамен во 2 семестре.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа студентов включает изучение лекционного материала вместе с соответствующими разделами печатных и электронных учебных пособий. Основной задачей самостоятельной работы является подготовка к устному докладу по разделу лекционного курса и решению задач практического занятия.

Оценивание уровня учебных достижений студента осуществляется в виде текущего контроля теоретических знаний (оценка устного доклада по разделу лекционного курса, тестирование по итогам изучения группы разделов) и промежуточной аттестации – зачета (решение и защита зачетной задачи с обсуждением использованных конструкций языка программирования) и экзамена (ответ на теоретические вопросы).

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю)

5.1. Описание шкал оценивания результатов обучения по дисциплине

Уровень сформированности компетенций (индикатора достижения компетенций)	Шкала оценивания сформированности компетенций						
	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	Не зачтено		зачтено				

<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущено много негрубых ошибок.	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько незначительных ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения Имели место грубые ошибки.	Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продemonстрированы все основные умения, решены все основные задачи с отдельными незначительными недочетами, выполнены все задания в полном объеме.	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие владения материалом. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки.	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продemonстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов.	Продemonстрирован творческий подход к решению нестандартных задач.

Шкала оценки при промежуточной аттестации

Оценка		Уровень подготовки
	Превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
	Отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»

Оценка		Уровень подготовки
зачтено	Очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	Хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
	Удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
незачтено	Неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
	Плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

5.2.1. Типовое контрольное задание, выносимое на зачет

Описать класс с именем WORKER, содержащий следующие элементы:

Скрытые поля:

- фамилия работника (char *),
- название занимаемой должности (char *),
- год поступления на работу (int).

Открытые методы:

- конструктор по умолчанию,
- конструктор с параметрами для заполнения всех полей класса,
- конструктор копирования,
- деструктор,
- операция присваивания,
- операция сравнения «меньше» (операция возвращает истину, если фамилия работника у первого операнда по алфавиту раньше, чем у второго),
- операция сравнения с числом «больше» (операция возвращает истину, если стаж работы превышает данное число)
- функция вывода на экран значений полей класса.

Память для строк – полей класса выделять динамически.

С помощью текстового редактора создать файл и записать в него данные для массива из десяти элементов типа **WORKER**.

Написать программу, выполняющую следующие действия:

- ввод из файла данных в динамический массив, состоящий из десяти элементов типа **WORKER**;
- упорядочить массив в алфавитном порядке по фамилиям работников.
- вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;
- если таких работников нет, вывести на дисплей соответствующее сообщение

5.2.2.Контрольные вопросы

№	Вопросы	Код формируемой компетенции
1.	Состав алгоритмического языка. Процедура создания исполняемой программы из исходного текста (ов) на C++.	ОПК-1, ОПК-3
2.	Алфавит языка C++. Идентификаторы. Ключевые слова. Константы. Комментарии.	
3.	Основные типы данных C++. Размер типа. Диапазон значений типа.	
4.	Структура программы на C++. Переменные. Общий вид объявления переменной. Область видимости и время жизни переменной.	
5.	Понятие «выражения». Правила преобразования типов операндов при вычислении выражения. Группы операций C++. Приоритет операций.	
6.	Базовые конструкции структурного программирования. Оператор «выражение» и операторы ветвления в C++.	
7.	Базовые конструкции структурного программирования. Операторы цикла в C++.	
8.	Базовые конструкции структурного программирования. Операторы передачи управления в C++.	
9.	Указатели. Способы инициализации указателей. Допустимые операции с указателями. Ссылки.	
10.	Массивы. Инициализация массива при объявлении. Многомерные массивы. Динамические массивы. Строки.	
11.	Типы данных, определяемые пользователем. Структуры, перечисления, объединения, битовые поля.	
12.	Понятие «функции». Правила объявления, определения и вызова функций.	
13.	Способы обмена данными между функциями. Глобальные переменные. Возвращаемое значение. Формальные и фактические параметры функции. Передача параметра по значению, по адресу и по ссылке. Передача массивов в качестве параметров. Параметры со значениями по умолчанию.	

14.	Рекурсивные функции. Плюсы и минусы использования рекурсии. Общий вид определения функции main. Назначение параметров и возвращаемого значения функции main.	
15.	Перегрузка функций. Неоднозначность при использовании перегруженной функции.	
16.	Шаблон функции. Правила определения и вызова функции-шаблона.	
17.	Директивы препроцессора. Включение файлов. Определение подстановки. Условная компиляция.	
18.	Описание класса. Описание объектов класса. Константные объекты и методы. Указатель this.	
19.	Конструкторы. Конструктор по умолчанию. Конструктор копирования. Деструктор.	
20.	Статические элементы класса. Дружественные функции и классы.	
21.	Перегрузка операций. Общие правила перегрузки унарных и бинарных операций.	
22.	Перегрузка операций. Перегрузка операции присваивания. Перегрузка операции приведения типа. Перегрузка операции индексирования.	
23.	Наследование. Ключи доступа к элементам базового класса. Порядок вызова конструкторов и деструкторов при создании (разрушении) объекта производного класса.	
24.	Виртуальные методы. Механизм позднего связывания. Абстрактные классы.	
25.	Шаблоны классов. Описание шаблона. Создание объектов шаблона.	
26.	Механизм обработки исключений в C++. Генерация и перехват исключений.	
27.	Пространства имён. Состав стандартной библиотеки C++.	
28.	Потоковые классы стандартной библиотеки C++. Стандартные потоки. Перегрузка операций извлечения из потока и включения в поток для типов, определенных пользователем.	
29.	Форматирование данных при вводе-выводе в поток. Флаги, форматирующие методы и манипуляторы.	
30.	Строковый класс стандартной библиотеки C++. Контейнерные классы. Типы контейнеров.	

5.2.3 Типовые контрольные задания, выносимые на экзамен

1. Написать программу, которая для целочисленного массива из N элементов определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

2. Написать программу, которая для одномерного массива произвольного числового типа определяет значение максимального и минимального элемента, а также среднее арифметическое значение элементов. Использовать шаблон функции, которая принимает массив в качестве входного параметра, возвращает минимум и максимум через выходные параметры, а среднее – через возвращаемое значение.
3. Создать шаблон класса «очередь». Данные добавляются в «хвост» очереди и извлекаются с «головы». В шаблоне реализовать конструктор, деструктор, метод добавления данных в очередь, метод извлечения данных из очереди. Перегрузить операцию вывода элементов очереди в стандартный поток ostream.

5.2.4. Типовые тестовые задания для оценки сформированности компетенций ОПК-1, ОПК-3

1. Выберите правильный ответ:

В алгоритмическом языке «словосочетанием» называется

- a. Лексема
- b. Оператор
- c. Директива
- d. Выражение

2. Установите правильную последовательность:

Для построения исполняемой программы из исходного текста на языке C++ последовательно вызываются:

- a. Компоновщик
- b. Препроцессор
- c. Компилятор

3. Выберите все правильные ответы:

К бинарным операциям C++ относится:

- a. инкремент
- b. сдвиг право
- c. логическое ИЛИ
- d. условная операция
- e. логическое отрицание
- f. сравнение на равенство

4. Выберите правильный ответ:

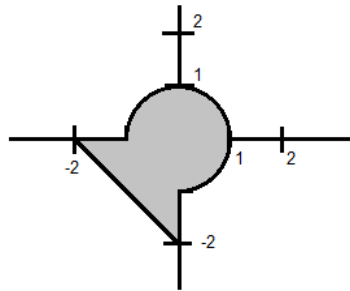
К операторам цикла с предусловием относятся операторы C++

- a. while и for
- b. for и do-while

с. do-while и while

5.2.5. Типовые задания/задачи для оценки сформированности компетенций ОПК-1, ОПК-3

1. Дана заштрихованная область (см. рисунок) и точка с координатами x и y . Написать программу, определяющую, попадает ли точка в область. Результат вывести в виде текстового сообщения.



2. С помощью генератора случайных чисел (`rand()`) сформировать одномерный массив из 20 элементов целого типа (значения в диапазоне от -100 до 100). Вывести массив на экран. Посчитать количество отрицательных и положительных элементов. Упорядочить массив по возрастанию методом выбора. Вывести результат на экран.
3. Создание простейшего класса комплексных чисел. Реализовать конструкторы (по умолчанию и с двумя параметрами для инициализации реальной и мнимой частей), операции для сложения и вычитания чисел, а также метод для вывода числа на экран. *Дополнительно:* переопределить операцию присваивания и убедиться, что операция присваивания, создаваемая компилятором по умолчанию, работает аналогично.
4. На основе реализации класса двусвязного списка, описанного в лекции, реализовать шаблон класса стека (в отличие от списка, добавлять и удалять значения можно только с одного конца). В классе реализовать конструктор, деструктор, функции добавления и удаления значений из стека. Перегрузить операцию вывода объекта стека в поток. Максимальное количество значений в стеке ограничено и задаётся параметром конструктора. Функции добавления и удаления значений должны выдавать сообщения об ошибке при переполнении стека или при попытке чтения из пустого стека. В `main` создать два стека для целых и плавающих чисел. В цикле добавить в стеки по 5-6 значений. Выводить на экран состояние стека на каждом шаге цикла. Повторить те же операции с удалением значений из стеков.
5. Обработка командных сообщений. Меню и панель инструментов. Реализовать простейшее приложение для Windows с использованием библиотеки классов MFC:
 - Класс главного окна сделать наследником `CFrameWnd`.
 - В редакторе ресурсов описать меню и панель инструментов. Команды кнопок панели инструментов должны соответствовать командам меню.

6. Приложение на основе диалога. Написать программу для Windows на основе диалога, переводящую время в часах, минутах и секундах в полное число секунд и обратно. Группа из 2-ух радиокнопок используется для выбора направления преобразования. Познакомится с мастерами создания MFC-проекта. Использовать Мастер классов для добавления переменных и функций-обработчиков.

6. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Огнева, М. В. Программирование на языке C++: практический курс: учебное пособие для бакалавриата и специалитета / М. В. Огнева, Е. В. Кудрина. — М.: Издательство Юрайт, 2017. — 335 с. — (Серия: Бакалавр и специалист). — ISBN 978-5-534-05123-0. — Режим доступа: www.biblio-online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4.
2. C/C++. Программирование на языке высокого уровня / Т.А. Павловская. — СПб: Питер, 2002. — 464 с., ил. 30 экз.
3. "Программирование в Win32 API на Visual Basic [Электронный ресурс] / Роман С.; Пер. с англ. - М.: ДМК Пресс, 2007. - (Серия "Для программистов")." — <http://www.studentlibrary.ru/book/ISBN5940741029.html>

б) дополнительная литература:

1. Б. Страуструп Язык программирования C++, 3-е изд./Пер. с англ. — СПб; М.: «Невский Диалект» - «Издательство БИНОМ», 1999 г. — 991 с., ил. 6 экз.
2. Фролов А.В., Фролов Г.В. Microsoft Visual C++ и MFC. Программирование для Windows 95 и Windows NT. — М.: ДИАЛОГ-МИФИ, 1996. — 288 с. — (Библиотека системного программиста; Т. 24) 1 экз.
3. Круглински Дэвид. Основы Visual C++/Пер с англ. — М.: Издательский отдел «Русская редакция» ТОО «Channel Trading Ltd.», 1997. — 696 с.: ил. 10 экз.

в) программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Пакет разработки приложений для операционной системы Windows Microsoft Visual Studio. <http://www.visualstudio.com>
2. Microsoft Developer Network Library. <http://msdn.microsoft.com/library>
3. Электронная библиотека книг братьев Фроловых. <http://www.frolov-lib.ru>

7. Материально-техническое обеспечение дисциплины

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: мультимедийным проектором и персональными компьютерами с установленной средой программирования Microsoft Visual Studio.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ВО ННГУ с учетом рекомендаций и ОПОП ВПО по направлению подготовки 09.03.02 «Информационные системы и технологии».

Автор (ы):

ст. преподаватель кафедры ИТФИ _____ Угольников А.Ю.

Рецензент

д.ф.-м.н., профессор, зав. каф.
статистической радиофизики и
мобильных систем связи РФФ _____ Мальцев А.А.

Заведующий кафедрой ИТФИ
д.т.н., профессор _____ Фидельман В.Р.

Программа одобрена на заседании методической комиссии физического факультета ННГУ
17июня 2020 года, протокол б/н.

Председатель УМК физ.ф-та _____ Перов А.А.