

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 президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

**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

---

г. Нижний Новгород

2024 год начала подготовки

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.О.06 Информатика относится к обязательной части образовательной программы.

## 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	Для текущего контроля успеваемости	Для промежуточной аттестации
ОПК-1: Способен применять естественнаучные и общеинженерные 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 ОПК-1.2: Уметь решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования ОПК-1.3: Иметь навыки теоретического и экспериментального исследования объектов профессиональной деятельности	ОПК-1.1: Знать 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++.  ОПК-1.2: Уметь логически верно описывать действия программы на алгоритмическом языке, создавать приложения для ОС Windows с использованием языка программирования C++, грамотно использовать операторы языка C++.  ОПК-1.3: Владеть средствами описания объектов и процессов реального мира языком программирования высокого уровня, инструментами разработки приложений, входящими в пакет Microsoft Visual Studio.	Задания  Тест	Экзамен: Контрольные вопросы Задачи  Зачёт: Задачи
ОПК-3: Способен решать	ОПК-3.1: Знать методы	ОПК-3.1:	Задания	

стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;	решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности 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 ОПК-3.3: Иметь навыки подготовки обзоров, аннотаций, составления рефератов, научных докладов	Знать основные конструкции структурного и объектно-ориентированного программирования языка C++, основные этапы создания 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.  ОПК-3.2: Уметь 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).  ОПК-3.3: Владеть мастерами автоматической генерации кода MFC, эффективными способами кодирования и документирования программы, методами решения функциональных и вычислительных задач	Тест	Экзамен: Контрольные вопросы Задачи  Зачёт: Задачи
ПК-2: Способен проводить научные исследования и управлять результатами научно-исследовательских работ при разработке и внедрении	ПК-2.1: Знать особенности проведения научных исследований ПК-2.2: Уметь обрабатывать и анализировать результаты научно-исследовательских работ ПК-2.3: Владеть навыками	ПК-2.1: Знать основные этапы создания программы на алгоритмическом языке (проектирование, кодирование, тестирование, отладка).	Задания Тест	Зачёт: Задачи  Экзамен: Контрольные вопросы Задачи

информационных технологий и систем	разработки и внедрения информационных технологий в системах научных исследований	<p>ПК-2.2: Уметь создавать алгоритмы для численного моделирования разнообразных объектов и систем, создавать собственные типы данных и наборы функций для взаимодействия с объектами этих типов.</p> <p>ПК-2.3: Владеть средствами описания объектов и процессов реального мира языком программирования высокого уровня.</p>		
------------------------------------	--	--	--	--

### 3. Структура и содержание дисциплины

#### 3.1 Трудоемкость дисциплины

	<b>очная</b>
<b>Общая трудоемкость, з.е.</b>	<b>7</b>
<b>Часов по учебному плану</b>	<b>252</b>
в том числе	
<b>аудиторные занятия (контактная работа):</b>	
- занятия лекционного типа	<b>48</b>
- занятия семинарского типа (практические занятия / лабораторные работы)	<b>64</b>
- КСР	<b>3</b>
<b>самостоятельная работа</b>	<b>101</b>
<b>Промежуточная аттестация</b>	<b>36</b> <b>Экзамен, Зачёт</b>

#### 3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
		0 ф 0	0 ф 0	0 ф 0	0 ф 0

1. Базовые средства языка C++.	38	8	8	16	22
2. Модульное программирование.	26	8	8	16	10
3. Объектно-ориентированное программирование.	38	8	8	16	22
4. Стандартная библиотека C++.	25	8	8	16	9
5. Введение в программирование для ОС Windows.	16	3	6	9	7
6. Обработка сообщений в MFC.	16	3	6	9	7
7. Диалоговые панели.	16	3	6	9	7
8. Вывод текста и графики в MFC.	16	3	6	9	7
9. Архитектура «Документ – Представление».	22	4	8	12	10
Аттестация	36				
КСР	3			3	
Итого	252	48	64	115	101

### Содержание разделов и тем дисциплины

1. Базовые средства языка C++. Алфавит языка. Основные типы данных C++. Структура программы на C++. Базовые конструкции структурного программирования. Типы данных, определяемые пользователем.
2. Модульное программирование. Функции. Способы взаимодействия функций. 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.
4. Стандартная библиотека C++. Потокные классы. Стандартные и файловые потоки. Строковый класс. Последовательные и ассоциативные контейнеры. Итераторы.
5. Введение в программирование для ОС Windows. История возникновения и развития ОС Windows. Модель программирования Windows. Типы данных Windows. Классы библиотеки MFC.
6. Обработка сообщений в MFC. Обработка сообщений в ОС Windows. Группы сообщений. Карта сообщений MFC.
7. Диалоговые панели. Типы диалоговых панелей. Редактирование и отображение диалоговой панели. Динамический обмен данными с диалоговой панелью.
8. Вывод текста и графики в MFC. Интерфейс графического устройства. Классы контекста устройства. Время жизни и состояние контекста устройства. Классы графических объектов.
9. Архитектура «Документ – Представление». Понятие документа и представления. Шаблон документа. Отношения между объектами архитектуры. Сериализация.

### 4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

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.

## **5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)**

### **5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:**

#### **5.1.1 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ОПК-1:**

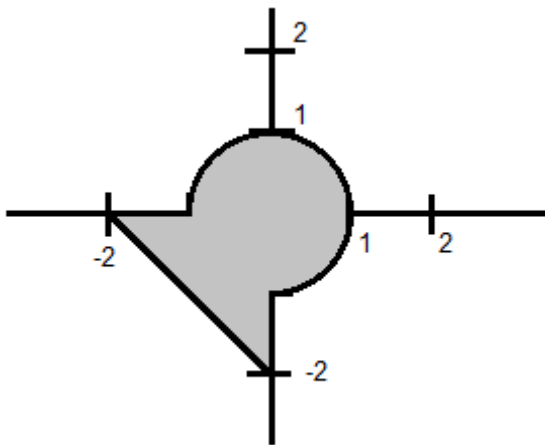
1. С помощью генератора случайных чисел (`rand()`) сформировать одномерный массив из 20 элементов целого типа (значения в диапазоне от -100 до 100). Вывести массив на экран. Посчитать количество отрицательных и положительных элементов. Упорядочить массив по возрастанию методом выбора. Вывести результат на экран.
2. 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.

#### **5.1.2 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ОПК-3:**

1. 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.
2. 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.

#### **5.1.3 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ПК-2:**

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. Описать структуру с именем STUDENT, содержащую следующие поля:

- а. Фамилия и инициалы (строка);
- б. Номер группы (целое число);
- с. Успеваемость (плавающее число).

Объявить массив структур из 5 элементов и инициализировать его. Упорядочить массив по возрастанию номера группы. Вывести на экран фамилии и номера групп тех студентов, успеваемость которых превышает 4.0. Сортировку организовать в виде отдельной функции.

#### Критерии оценивания (оценочное средство - Задания)

Оценка	Критерии оценивания
зачтено	Программа реализует поставленные функциональные задачи.
не зачтено	Программа не реализована или работает с ошибками.

#### 5.1.4 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ОПК-1:

1. Выберите правильный ответ:

*В алгоритмическом языке «словосочетанием» называется*

1. Лексема
2. Оператор
3. Директива
4. Выражение

2. Установите правильную последовательность:

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

1. Компоновщик
2. Препроцессор
3. Компилятор

### 5.1.5 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ОПК-3:

1. Выберите все правильные ответы:

*К бинарным операциям C++ относится:*

1. инкремент
2. сдвиг право
3. логическое ИЛИ
4. условная операция
5. логическое отрицание
6. сравнение на равенство

2. Выберите правильный ответ:

*К операторам цикла с предусловием относятся операторы C++*

1. while и for
2. for и do-while
3. do-while и while

### 5.1.6 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-2:

1. Установите соответствие:

- |           |               |
|-----------|---------------|
| 1. struct | 1. операция   |
| 2. define | 2. тип данных |
| 3. switch | 3. директива  |
| 4. new    | 4. оператор   |

1. Выберите правильный ответ:

*Диапазон значений (0...255)10 соответствует типу данных*

1. unsigned int
2. unsigned char
3. long double
4. signed char

### Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
зачтено	Решено 40% или более заданий
не зачтено	Решено менее 40% заданий



## 5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

### Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения 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 несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	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 отдельным и несущественными 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 базовых навыков. 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 некоторым и недочетами	Продemonстрированы базовые навыки при решении стандартных задач с некоторым и 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 оценивания при промежуточной аттестации

Оценка	Уровень подготовки
--------	--------------------

<b>зачтено</b>	<b>превосходно</b>	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	<b>отлично</b>	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	<b>очень хорошо</b>	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	<b>хорошо</b>	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	<b>удовлетворительно</b>	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»
<b>не зачтено</b>	<b>неудовлетворительно</b>	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	<b>плохо</b>	Хотя бы одна компетенция сформирована на уровне «плохо»

### 5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

#### 5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-1

1. Состав алгоритмического языка. Процедура создания исполняемой программы из исходного текста (ов) на C++.
2. Алфавит языка C++. Идентификаторы. Ключевые слова. Константы. Комментарии.
3. Основные типы данных C++. Размер типа. Диапазон значений типа.

#### 5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-3

1. Базовые конструкции структурного программирования. Операторы передачи управления в C++.
2. Указатели. Способы инициализации указателей. Допустимые операции с указателями. Ссылки.
3. Массивы. Инициализация массива при объявлении. Многомерные массивы. Динамические массивы. Строки.

#### 5.3.3 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-2

1. Наследование. Ключи доступа к элементам базового класса. Порядок вызова конструкторов и деструкторов при создании (разрушении) объекта производного класса.
2. Механизм обработки исключений в C++. Генерация и перехват исключений

#### Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
превосходно	Обучаемый ответил на вопросы билета в полном объёме, ответил на дополнительный вопрос повышенной трудности.
отлично	Обучаемый ответил на вопросы билета в полном объёме, ответил на дополнительный вопрос.
очень хорошо	Обучаемый ответил на вопросы билета в полном объёме, но не ответил на дополнительный вопрос.
хорошо	Обучаемый ответил на вопросы билета с достаточной полнотой, продемонстрировал способность обоснования ответов с помощью преподавателя.
удовлетворительно	Обучаемый изложил, но недостаточно обосновал ответы на основные положения вопросов билета.
неудовлетворительно	Обучаемый не смог внятно изложить основные положения вопросов билета.
плохо	Обучаемый не ответил на вопросы билета.

#### **5.3.4 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ОПК-1**

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.

#### **5.3.5 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ОПК-3**

1. Написать программу, которая для вещественного массива из N элементов определяет сумму его элементов, расположенных правее последнего отрицательного элемента.
2. 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.

#### **5.3.6 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-2**

1. Даны два массива из N целых чисел каждый. Определить, для какого из них среднее арифметическое значение элементов больше. Для решения задачи написать функцию, принимающую массив в качестве параметра и возвращающую среднее значение.

2. Реализовать рекурсивную функцию быстрой сортировки массива целых чисел.

### Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
превосходно	Обучаемый самостоятельно и правильно решил задачу билета.
отлично	Обучаемый самостоятельно и правильно решил задачу билета.
очень хорошо	Обучаемый самостоятельно и правильно решил задачу билета.
хорошо	Обучаемый самостоятельно и правильно решил задачу билета.
удовлетворительно	Обучаемый применял правильные методы для решения задачи билета.
неудовлетворительно	Обучаемый не смог применить правильные методы для решения задачи билета.
плохо	Обучаемый не продемонстрировал методы решения задачи билета.

### 5.3.7 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ОПК-1

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

### 5.3.8 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ОПК-3

Описать класс с именем TRAIN, содержащий следующие элементы:

- Скрытые поля:
  - название пункта назначения (char \*),
  - номер поезда (int),
  - время отправления (double в формате «чч.мм»).
- 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 название пункта назначения у первого операнда по алфавиту раньше, чем у второго),
  - операция сравнения с числом «больше» (операция возвращает истину, если время отправления больше данного числа)
  - 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 восьми элементов типа TRAIN.

Написать программу, выполняющую следующие действия:

- ввод из файла данных в динамический массив, состоящий из восьми элементов типа TRAIN;
- упорядочить массив в алфавитном порядке по названиям пунктов назначения;
- вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени;
- если таких поездов нет, выдать на дисплей соответствующее сообщение.

### 5.3.9 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-2

Описать класс с именем NOTE, содержащий следующие элементы:

- Скрытые поля:
  - фамилия (char \*),
  - номер телефона (int),
  - день рождения (int[3] в формате «int[0] = ДД, int[1] = ММ, int[2] = ГГГГ»).
- 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 дата дня рождения у первого операнда раньше, чем у второго),
- операция сравнения на равенство с числом (операция возвращает истину, если номер телефона совпадает с данным числом)
- 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 восьми элементов типа NOTE.

Написать программу, выполняющую следующие действия:

- ввод из файла данных в динамический массив, состоящий из восьми элементов типа NOTE;
- упорядочить массив по датам дней рождения;
- вывод на экран информации о человеке, номер телефона которого введен с клавиатуры;
- если такого нет, выдать на дисплей соответствующее сообщение.

### Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
зачтено	Программа реализует поставленные функциональные задачи. Обучаемый свободно ориентируется в исходном коде программы, способен внести в код незначительные модификации.
не зачтено	Программа не реализована или работает с ошибками. Обучаемый не в состоянии определить место и причину ошибки в исходном коде программы.

## 6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Павловская Татьяна Александровна. С/С++. Программирование на языке высокого уровня : для магистров и бакалавров. - СПб. : Питер, 2013. - 461 с. - (Учебник для вузов). - ISBN 978-5-496-00031-4 : 324.00., 1 экз.
2. Круглински Д. Дж. Программирование на Microsoft Visual C++ 6.0 для профессионалов = Programming Microsoft Visual C++ 6.0 : [пер. с англ.]. - СПб. [и др.] : Питер : Русская редакция, 2001. - 864 с. : ил. + 1 электрон. опт. диск (CD-ROM). - ISBN 5-272-00385-3. - ISBN 1-57231-857-0 (англ.) : 458.00., 3 экз.

Дополнительная литература:

1. Шеферд Дж. Программирование на Microsoft Visual C ++ .NET : пер. с англ. - 2-е изд. - М. ; СПб. : Русская редакция : Питер, 2007. - 928 с. + 1 электрон. опт. диск (CD-ROM). - (Мастер-класс). - ISBN 5-469-01178-X ("Питер") : 482.00., 5 экз.
2. Угольников Александр Юрьевич. Отладка приложений в Microsoft Visual Studio : учеб.-метод.

пособие для студентов ННГУ, обучающихся по направлению подготовки 230400

"Информационные системы и технологии" / А. Ю. Угольников ; Нижегород. гос. ун-т им. Н. И. Лобачевского. - Нижний Новгород : Нижегород. гос. ун-т, 2012. - 44 с., 40 экз.

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 (модуля)**

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

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 ННГУ по направлению подготовки/специальности 09.03.02 - Информационные системы и технологии.

Автор(ы): Угольников Александр Юрьевич.

Рецензент(ы): Мальцев Александр Александрович, доктор физико-математических наук.

Заведующий кафедрой: Морозов Олег Александрович, доктор физико-математических наук.

Программа одобрена на заседании методической комиссии от 09.01.2024, протокол № б/н.