

**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное учреждение**  
**высшего профессионального образования**  
**«Череповецкий государственный университет»**

**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

---

институт (факультет)

**МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭВМ**

---

кафедра

**УТВЕРЖДЕНО**

на заседании ученого совета института

28.08. 2015 г.,

протокол № 1

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Наименование дисциплины:**

**ПРОГРАММИРОВАНИЕ**

---

(полное наименование дисциплины (модуля) в соответствии с учебным планом)

**Уровень профессионального образования, в рамках которого изучается  
дисциплина (модуль):**

**бакалавриат**

---

(высшее образование – бакалавриат; высшее образование – специалитет, магистратура;  
высшее образование – подготовка кадров высшей квалификации по программам подготовки научно-  
педагогических кадров в аспирантуре)

**Направления подготовки (специальности):**

**09.03.04 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

---

(коды и наименования укрупненных групп направлений подготовки (специальностей) / коды и наименования  
направлений подготовки (специальностей), в рамках которых изучается дисциплина (модуль), в соответствии с  
перечнем специальностей и направлений подготовки высшего образования, утверждаемым Министерством  
образования и науки Российской Федерации)

## Общие сведения о программе

Рабочая программа дисциплины (модуля) составлена на основе требований ФГОС ВО по направлениям подготовки (специальностям):

Код и наименование направления подготовки (специальности)	Дата и номер приказа Минобрнауки России
09.03.04 Программная инженерия	12.03.2015, № 229

Рабочая программа дисциплины (модуля) рассмотрена и рекомендована к утверждению на заседании кафедры

### МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭВМ

---

наименование кафедры

от 01.07.2015, протокол № 11.

Сведения о разработчике(ах) программы:

ЕРШОВ ЕВГЕНИЙ ВАЛЕНТИНОВИЧ Д.Т.Н., ПРОФЕССОР, ЗАВЕДУЮЩИЙ  
КАФЕДРОЙ МПО ЭВМ

---

(ФИО, ученая степень, ученое звание, должность, место работы)

СЕЛИВАНОВСКИХ ВЕРА ВИТАЛЬЕВНА К.Т.Н, ДОЦЕНТ КАФЕДРЫ МПО  
ЭВМ

---

(ФИО, ученая степень, ученое звание, должность, место работы)

---

## Содержание

- 1** Организационно-методический раздел
- 2** Структура и содержание дисциплины (модуля)
- 3** Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)
- 4** Методические указания для обучающихся по освоению дисциплины (модуля)
- 5** Учебно-методическое обеспечение дисциплины (модуля)
- 6** Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)
- 7** Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

# 1 ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

## 1.1 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Программирование» относится к одной из основных дисциплин, формирующих профессиональные компетенции, и состоит из трех модулей: «Основы программирования», «Структурное программирование», «Объектно-ориентированное программирование».

Необходимыми условиями для освоения дисциплины являются: знание информационно-коммуникационных технологий в пределах общего (полного) среднего образования, умение использовать способы алгоритмизации для разработки несложных алгоритмов, владение базовыми навыками работы на персональном компьютере.

Содержание дисциплины является логическим продолжением содержания школьного курса информатики и формирует основу для освоения дисциплин «Структуры и алгоритмы обработки данных», «Профильное программирование», «Технология разработки программного обеспечения», «Проектирование программно-информационных систем», а также других дисциплин, связанных с использованием методов и языков программирования.

## 1.2 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Перечень планируемых результатов освоения образовательной программы – компетенции, формируемые в результате освоения дисциплины (модуля)	Перечень планируемых результатов обучения по дисциплине
--	---

<p>Частично формируются обще- профессиональные и профессиональные компетенции:</p> <p>- готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-3);</p> <hr/> <p>- готовность применять основные методы и инструменты разработки программного обеспечения (ПК-1);</p> <hr/> <p>- способность к формализации в своей предметной области с учетом ограничений используемых методов исследования (ПК- 12);</p> <hr/> <p>- готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности (ПК-13);</p> <hr/> <p>- готовность обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности (ПК-14);</p>	<p><b>Умение</b> использовать технологии структурного, модульного и объектно- ориентированного программирования для создания кода, тестировать, документировать и получить решение поставленных задач различной сложности и объема. <b>Владение</b> языками процедурного и объектно-ориентированного программирования;</p> <hr/> <p><b>Умение</b> применять основные методы и инструменты разработки программного обеспечения к решению практических задач.</p> <hr/> <p><b>Владение навыками</b> формализованного представления информации и процесса обработки данных в виде программы на языке программирования.</p> <hr/> <p><b>Знание</b> на уровне представлений: современные технические и программные средства взаимодействия с компьютером, основные концепции структурного и объектно-ориентированного программирования; на уровне воспроизведения: современные технологии разработки алгоритмов и программ, методы тестирования, отладки и испытания программ, на уровне понимания: технологии структурного, модульного и объектно- ориентированного программирования; <b>Умение</b> использовать технологии структурного, модульного и объектно- ориентированного программирования для решения практических задач.</p> <hr/> <p><b>Умение</b> обосновать выбор алгоритма решения прикладной задачи; <b>Владение</b> навыками самостоятельной разработки алгоритма решения прикладной задачи;</p>
--	--

## 2 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

**2.1 Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся**

### 09.03.04-П1-14

#### Модуль Основы программирования

коды и наименования направлений подготовки (специальностей) / укрупненных групп направлений подготовки и специальностей													
Форма обучения	Кол-во зачетных единиц, всего	Кол-во часов, всего	в т.ч. аудиторная работа				в т.ч. самостоятельная работа						Форма промежуточной аттестации  (Экзамен / Зачет)
			Всего	из них:			Всего	из них:					
				Л	ЛР	ПЗ		КР / КП	РГЗ	Кр	Р	Д	
очная	5	180	40	16	24	0	140	0	0	20	0	120	Зачет

#### Модуль Структурное программирование

коды и наименования направлений подготовки (специальностей) / укрупненных групп направлений подготовки и специальностей													
Форма обучения	Кол-во зачетных единиц, всего	Кол-во часов, всего	в т.ч. аудиторная работа				в т.ч. самостоятельная работа						Форма промежуточной аттестации  (Экзамен / Зачет)
			Всего	из них:			Всего	из них:					
				Л	ЛР	ПЗ		КР / КП	РГЗ	Кр	Р	Д	
очная	10	360	80	32	48	0	280	180	0	0	0	100	Зачет, Экзамен

## Модуль Объектно-ориентированное программирование

коды и наименования направлений подготовки (специальностей) / укрупненных групп направлений подготовки и специальностей													
Форма обучения	Кол-во зачетных единиц, всего	Кол-во часов, всего	в т.ч. аудиторная работа				в т.ч. самостоятельная работа						Форма промежуточной аттестации  (Экзамен / Зачет)
			Всего	из них:			Всего	из них:					
				Л	ЛР	ПЗ		КР / КП	РГЗ	Кр	Р	Д	
очная	8	288	72	32	40	0	216	150	0	0	0	66	Экзамен, Экзамен

*Виды учебной работы: Курсовая работа (проект) – КР/КП; Расчетно-графические задания - РГЗ; Контрольные работы - Кр; Реферат - Р; Другие виды самостоятельной работы - Д.*

*Заполняется для каждого направления подготовки (специальности) или для группы направлений подготовки (специальностей). В случае, если программа разработана для одного направления подготовки (специальности), то его можно не указывать.*

## 2.2 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

№ и наименование раздела (модуля) дисциплины	Краткое содержание
<b>Модуль Основы программирования</b>	
1 Основы алгоритмизации.	Алгоритмы, способы записи алгоритмов. Требования к алгоритмам. Типы вычислительных процессов и типы алгоритмов. Требования ГОСТ к оформлению и проектированию алгоритмов.
2. Введение в программирование	Программирование: сущность и назначение. Жизненный цикл программного продукта. Принципы структурного и модульного программирования. Этапы разработки программного обеспечения. Критерии качества программного обеспечения. Классификация языков программирования. Синтаксис и семантика. Средства описания языков программирования. Метаязык.
3. Основы программирования на языке C++	Синтаксические элементы программы: комментарии, ключевые слова, идентификаторы, константы, знаки пунктуации. Простые собственные типы данных. Простые пользовательские типы данных. Объявление, определение, инициализация переменных. Иерархия типов. Совместимость и преобразование типов. Операции и выражения. Программирование ввода/вывода. Структура программы на C++. Операторы пустой, составной, вызова функции. Операторы ветвления. Операторы цикла.
<b>Модуль Структурное программирование</b>	
4. Структурное программирование на C++	Указатели. Объявление указателей. Операции над указателями. Динамические переменные. Указатели и массивы. Связь указателей и массивов. Динамические массивы. Структуры. Объединения. Функции. Передача параметров функции. Передача массивов в качестве параметров функции. Перегрузка функций. Текстовые файлы в C++. Файловые потоки ввода-вывода. Создание потока ввода-вывода. Открытие и закрытие файла. Обработка исключительных ситуаций. Чтение-запись символов в файл. Ввод-вывод с преобразованием типов. Чтение символьных строк из потока ввода.
5. Общие сведения о системах программирования Eclipse и MS Visual Studio	Интерфейс. Палитра компонентов. Инспектор объектов. Компиляция и выполнение программы. Консольное и визуальное приложение.
6. Анализ сложности алгоритмов	Оценка сложности алгоритмов. Определение времени выполнения алгоритмов, использующих рекурсию. Основы доказательства правильности алгоритмов.
<b>Модуль Объектно-ориентированное программирование</b>	



№ и наименование раздела (модуля) дисциплины	Краткое содержание
7. Основные идеи и концепции ООП	<p>Использование объекта в качестве основной компоненты программы и децентрализация управления, реализуемая представлением программы совокупностью взаимодействующих объектов. Моделирование поведения объектов реального мира, наглядное представление объектов, создание и использование АД, производные АД.</p> <p>Инкапсуляция и расширяемость типов. Семантика объекта: данные как отражение состояния объекта и методы как средства обеспечения взаимодействия объектов. Представление данных. Классификация методов: конструкторы, деструкторы, селекторы, модификаторы. Классы объектов: назначение и семантика, классы и абстрактные типы данных, наглядное представление классов. Объекты как экземпляры классов. Основные действия с объектами: создание, инициализация, обработка, разрушение. Конструкторы и деструкторы, основное назначение. Перегрузка операторов и функций. Ключевые слова: friend, operator. Шаблоны и обобщенное программирование. Ключевое слово: template. Контейнерные классы. Основные составляющие STL: контейнеры, итераторы, алгоритмы. Наследование: базовый и производный класс, иерархия классов. Полиморфизм. Абстрактный базовый класс. Обработка исключительных ситуаций, проверка утверждений. Преимущества и недостатки ООП. Основные свойства языка ООП. Место языка C++ во внедрении и развитии средств ООП. Значение языка C++ для профессиональной подготовки программистов.</p>
8. Абстрактные типы данных - классы	<p>Классы: компонентные данные, компонентные функции, способы задания доступа, синтаксическая структура определения класса, объявление, определение и инициализация объектов, объекты, объявленные как константы. Область видимости класса: оператор разрешения области видимости (: :) - унарная форма, бинарная форма; вложенные классы - классы, вложенные в классы, классы, вложенные в блоки, вложенные определения функций. Компонентные данные: назначение, основные свойства, синтаксис и семантика определений компонентных данных, статические компонентные данные, статические константы, изменяемые компонентные данные (mutable). Компонентные функции: назначение, основные свойства, доступ к компонентным функциям, доступ компонентных функций к компонентным данным - функции селекторы, функции модификаторы. Встроенные компонентные функции: назначение, синтаксис и семантика определения, рекомендации по использованию; компонентные функции, определяемые вне класса: назначение, синтаксис и семантика определения и объявления, рекомендации по использованию. Указатель this, компонентные функции типа static и const (статические и постоянные компонентные функции): назначение, синтаксис и семантика определения и объявления, рекомендации по использованию. Тестирование класса.</p>

№ и наименование раздела (модуля) дисциплины	Краткое содержание
9. Создание и уничтожение объектов	<p>Конструкторы: назначение, наличие нескольких конструкторов у одного класса - причины, разделение функций, способы задания конструктора: по умолчанию или явное. Явное описание (определение или объявление) конструктора, размещение определения в теле класса и влияние размещения на функции конструктора, размещение определения вне своего класса, параметры конструктора и возможности их использования. Конструкторы для инициализации объектов, инициализаторы конструктора; конструкторы для преобразования типов, ключевое слово <code>explicit</code>. Конструкторы умолчания: назначение, способы задания, правила описания. Конструкторы копирования: назначение, способы задания, правила описания. Вызов конструктора: назначение, способы реализации, задание аргументов; явный и автоматический вызовы: ситуации использования, рекомендации по применению. Деструкторы: назначение, основные свойства, способы задания: по умолчанию или явное. Явное описание (определение или объявление) деструктора, размещение определения в теле класса и влияние размещения на функции деструктора, размещение определения вне своего класса, параметры деструктора и возможности их использования. Вызов деструктора: назначение, способы реализации, явный и автоматический вызовы: ситуации использования, рекомендации по применению. Отладка и тестирование системы конструкторов и деструкторов.</p>
10. Специальный полиморфизм	<p>Преобразование АТД к встроенному типу. Перегрузка функций, алгоритм выбора перегруженной функции. Дружественные функции: основные свойства, номенклатура, необходимость, полезность и опасность применения, наборы дружественных функций, дружественные классы, объявление – размещение, синтаксис, семантика, требования к параметрам и типу возвращаемого значения; определение – размещение, область видимости; вызов функции. Перегрузка операторов: синтаксис, семантика, приоритет и порядок выполнения, доступные и недоступные для перегрузки операторы, перегружаемые как обычные функции, только как компонентные функции. Перегрузка унарных и бинарных операторов: синтаксис и семантика. Перегрузка операторов присваивания и индексирования: синтаксис и семантика. Перегрузка операторов ввода-вывода (<code>&lt;&lt;</code>, <code>&gt;&gt;</code>): синтаксис и семантика. Перегрузка оператора <code>()</code> для индексирования: синтаксис и семантика. Перегрузка операторов доступа к элементу класса через указатель на объект <code>→</code>, через указатель на элемент <code>→*</code>: синтаксис и семантика. Перегрузка операторов <code>new</code> и <code>delete</code>: синтаксис и семантика.</p>

№ и наименование раздела (модуля) дисциплины	Краткое содержание
11. Параметрический полиморфизм	Шаблонные классы и шаблонные функции: назначение, параметризованные типы данных, синтаксис и семантика. Шаблоны классов: назначение, синтаксис и семантика, объявление и определение шаблона, дружественные шаблонному классу функции, статические элементы шаблонного класса, аргументы шаблона класса. Шаблоны функций: назначение, синтаксис и семантика, объявление, определение, вызов и перегрузка шаблонных функций, аргументы шаблона функции.
12. Наследование и иерархия классов	Основные элементы методологии объектно-ориентированного проектирования, реализация механизма повторного использования кода, полиморфная обработка родственных объектов – чистый полиморфизм. Отношение наследования: определение, семантика, базовый и производный классы, доступ к элементам классов, преимущества использования производных классов. Построение производного класса специализацией базового, построение базового класса обобщением производного, иерархия классов. Правила преобразования типов и видимость в иерархии родственных классов; конструкторы и деструкторы производного и базового классов: определение, вызов. Наследование интерфейса, наследование реализации: назначение, особенности применения, достоинства и недостатки. Виртуальные функции: назначение, определение, семантика, вызов; чисто виртуальные функции - отложенные методы. Абстрактные базовые классы: назначение, определение, семантика, применение. Шаблоны и наследование: достоинства и недостатки каждого механизма, формы комбинирования и особенности их применения. Множественное наследование: причины появления и назначение, определение отношения множественного наследования - синтаксис и семантика; виртуальное наследование, виртуальные функции при множественном наследовании, порядок выполнения конструкторов, практическое использование множественного наследования. Тестирование системы классов связанных отношением наследования.

№ и наименование раздела (модуля) дисциплины	Краткое содержание
13. Обработка исключений	<p>Проверка утверждений (assert.h), сигналы - асинхронные исключения (signal.h). Механизм обработки исключений: номенклатура ситуаций и модели управления ими, средства поддержки управления в языке C++. Запуск исключений, перезапуск исключений, выражения исключений: синтаксис и семантика, спецификация исключения. Пробные блоки try, обработчики исключений catch: синтаксис и семантика, иерархии классов для управления исключительными ситуациями. Обработчики terminate() и unexpected(): назначение, особенности применения. Использование стандартных исключений: отсутствие свободной памяти, тестирование приведений типов, логические ошибки, ошибки времени выполнения. Обработка исключений как механизм восстановления после сбоев и как механизм передачи управления, модель завершения, используемая в C++.</p>
14. Итераторы, стандартная библиотека шаблонов (STL)	<p>Итераторы: назначение, синтаксис и семантика, объявления, определение, вызов, особенности применения. Последовательные и ассоциативные контейнеры, интерфейсы контейнеров, адаптеры контейнеров. Итераторы ввода, итераторы вывода, итераторы прохода вперед, двусторонние итераторы, итераторы произвольного доступа, итераторы istream_iterator и ostream_iterator, адаптеры итераторов. Алгоритмы сортировки, алгоритмы, неизменяющие последовательность содержимого контейнера, алгоритмы, изменяющие последовательность содержимого контейнеров, численные алгоритмы. Арифметические, сравнивающие, логические объекты-функции, адаптеры объектов-функций.</p>

№ и наименование раздела (модуля) дисциплины	Краткое содержание
15. Разработка объектно-ориентированных программ	Конструирование объектно-ориентированных программ: структура ООП - разделение определений классов и определений компонентных функций; области видимости и время существования переменных - локальные, статические и глобальные переменные; организация файловой структуры программы для эффективной разработки, отладки и модификации. Конструирование иерархий классов: принципы построения, рекомендации, примеры; создание и использование объектов во время исполнения программы: средства программирования и рекомендации по их использованию, организация массивов объектов; использование конструкторов, вызывающих другие конструкторы. Идентификация типа на этапе выполнения: назначение, операторы - <code>dynamic_cast</code> , <code>typeid</code> , класс <code>type_info</code> . Основные особенности использования функций, связанные с наследованием. Методика тестирования объектно-ориентированных программ. Объектно-ориентированное проектирование: CRC-карточки, диаграммы Вассермана-Пирчера, унифицированный язык моделирования (UML). Принципы и штампы объектно-ориентированного проектирования, тенденции развития методов, языков и инструментальных средств ООП, автоматизация разработки программ, визуализация ООП.

## Лекции

№ раздела (модуля)	Тема лекции	Количество часов
<b>Модуль Основы программирования</b>		
1	Этапы решения задач на ЭВМ. Проектирование алгоритмов	2
2	Принципы структурного программирования. Классификация языков программирования. Средства описания языков программирования.	2
3	Состав языка C++. Типы данных C++.	4
3	Операции и выражения. Стандартные математические функции.	2
3	Структура программы. Программирование ввода/вывода.	2
3	Операторы языка C++	4
<b>Модуль Структурное программирование</b>		
4	Указатели. Операции над указателями.	2
4	Массивы. Связь массивов и указателей. Динамические массивы	4
4	Строковый тип данных.	4
4	Структуры и объединения.	4
4	Функции. Передача параметров. Перегрузка функций.	4
4	Программирование доступа к файлам. Текстовые и бинарные файлы.	4
5	Системы программирования Eclipse и MS Visual Studio.	4

№ раздела (модуля)	Тема лекции	Количество часов
6	Оценка сложности алгоритмов.	4
6	Основы доказательства правильности алгоритмов.	2
<b>Модуль Объектно-ориентированное программирование</b>		
7	Основные идеи и концепции ООП	2
8	Классы, инкапсуляция и расширяемость типов	2
8	Компонентные данные и функции, объекты и их поведение	2
9	Создание объектов, конструкторы	4
9	Уничтожение объектов, деструкторы	2
10	Перегрузка преобразований типов данных, перегрузка функций, дружественные функции	2
10	Перегрузка операторов	2
11	Шаблонные классы	2
11	Шаблонные функции	2
12	Наследование интерфейса, иерархия родственных типов	2
12	Наследование реализации, повторное использование кода	2
12	Полиморфная обработка родственных объектов, интерфейсы	2
13	Проверка утверждений, асинхронные исключения	1
13	Механизм обработки исключений	1
14	Итераторы, стандартная библиотека шаблонов (STL)	2
15	Разработка объектно-ориентированных программ	2

### Лабораторные работы

№ раздела (модуля)	Тема лабораторной работы	Количество часов
<b>Модуль: Основы программирования</b>		
3	Программирование ввода/вывода в консольном приложении.	4
3	Стандартные математические функции.	4
3	Программирование линейных алгоритмов.	4
3	Операторы ветвления.	4
3	Операторы цикла.	8
<b>Модуль: Структурное программирование</b>		
5	Основные принципы работы в визуальной системе программирования Microsoft Visual Studio. Программирование указателей.	4
4	Обработка одномерных массивов.	4
4	Обработка двумерных массивов.	4
5	Основные принципы работы в визуальной системе программирования Microsoft Visual Studio. Обработка массивов.	4
4	Обработка динамических массивов	4
4	Обработка данных строкового типа.	4
4	Структуры и объединения.	4
4	Функции. Механизм передачи параметров	8
4	Перегрузка функций	4
4	Программирование доступа к файлам.	4
6	Оценка сложности алгоритмов.	4

№ раздела (модуля)	Тема лабораторной работы	Количество часов
<b>Модуль: Объектно-ориентированное программирование</b>		
8	Разработка абстрактных типов данных	4
9	Создание и уничтожение объектов класса	4
10	Перегрузка операторов	4
10	Дружественные функции	4
11	Программирование шаблонных классов и функций	4
12	Программная реализация отношения простого наследования	4
12	Программная реализация отношения множественного наследования	4
13	Программирование обработки исключительных ситуаций	4
14	Проектирование взаимодействия объектов классов	4
15	Проектирование объектно-ориентированных программ	4

### 3 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

#### 3.1 Перечень компетенций с указанием этапов их формирования

Компетенции	Оценочные средства
готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-3)	<ol style="list-style-type: none"> <li>1. Выполнение лабораторных работ, защита отчетов по лабораторным работам</li> <li>2. Контрольная работа</li> <li>3. Выполнение и защита курсовой работы</li> </ol>
готовность применять основные методы и инструменты разработки программного обеспечения (ПК-1)	<ol style="list-style-type: none"> <li>1. Выполнение лабораторных работ, защита отчетов по лабораторным работам</li> <li>2. Выполнение и защита курсовой работы</li> <li>3. Тест</li> <li>4. Зачет</li> <li>5. Экзамен</li> </ol>
способность к формализации в своей предметной области с учетом ограничений используемых методов исследования (ПК-12)	<ol style="list-style-type: none"> <li>1. Выполнение лабораторных работ, защита отчетов по лабораторным работам</li> <li>2. Выполнение и защита курсовой работы</li> <li>3. Экзамен</li> </ol>
готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности (ПК-13)	<ol style="list-style-type: none"> <li>1. Выполнение и защита курсовой работы</li> </ol>
готовность обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности (ПК-14)	<ol style="list-style-type: none"> <li>1. Выполнение и защита курсовой работы</li> </ol>

### 3.2 Типовые контрольные задания и иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

#### Модуль Основы программирования

##### 1) Выполнение и защита лабораторной работы «Операторы цикла».

1. Среди трехзначных чисел найти и вывести те, квадрат которых оканчивается тремя цифрами, составляющими исходное число.
2. Вычислите значение выражения. Значение  $x$  вводится с клавиатуры.

$$Y = \sum_{n=1}^{10} \left( x^{n-1} + \frac{x^n}{2^n} \right)$$

2) Тест состоит из 17 вопросов, дающих в сумме 10 баллов.

Некоторые вопросы теста:

Вопрос 1 ▾ Выбор варианта ответа (единственный верный) - 0.2

Идентификатор - это:

- ☒ А. Последовательность букв, цифр и символа подчеркивания;
- ☐ Б. Последовательность букв;
- ☐ В. Последовательность букв, цифр и знаков операций;
- ☐ Г. Последовательность слов, букв, чисел;

Правильный ответ: А

Вставить новый вопрос

Вопрос 2 ▾ Выбор варианта ответа (единственный верный) - 0.2

Переменная - это:

- ☒ А. Объект, способный принимать различные значения;
- ☐ Б. Значения чисел;
- ☐ В. Меняющееся число;
- ☐ Г. Динамический объект;

Правильный ответ: А

Вставить новый вопрос

Вопрос 3 ▾ Выбор варианта ответа (единственный верный) - 0.2

Идентификатором является:

- ☒ А. Name2;
- ☐ Б. Name-“Петя”;
- ☐ В. /Name;
- ☐ Г. \*Name1;



3. Контрольная работа на тему «Разработка алгоритмов для различных типов вычислительных процессов». Типовые задания:

Разработать алгоритм для решения задачи. Найти произведение цифр четырехзначного числа, введенного пользователем.

Разработать алгоритм для решения задачи. Определить, есть ли среди цифр трехзначного числа, введенного пользователем, одинаковые цифры.  
Разработать алгоритм для решения задачи. Для произвольных вещественных чисел  $a$ ,  $b$  и  $c$  найти корни квадратного уравнения  $ax^2 + bx + c = 0$ .

### ВОПРОСЫ К ЗАЧЕТУ

1.	Критерии качества программы
2.	Жизненный цикл программного продукта
3.	Понятие алгоритма. Требования к алгоритму
4.	Основные способы записи алгоритмов
5.	Объявление, определение, инициализация переменных в C++
6.	Файловая структура программы в C++
7.	Организация компиляции и выполнения программы в C++ Builder и MS Visual Studio
8.	Ввод/вывод информации в C++
9.	Простые собственные типы данных в C++
10.	Простые пользовательские типы данных в C++
11.	Операторы присваивания, сравнения, логические, побитовые в C++
12.	Операторы пустой, составной, вызова функции в C++
13.	Операторы ветвления в C++
14.	Операторы цикла в C++
15.	Операторы передачи управления в C++

### Модуль Структурное программирование

#### 1) Выполнение и защита лабораторной работы «Перегрузка функций».

Задание: Разработать программное обеспечение для нахождения максимального элемента в одномерном динамическом массиве, используя перегрузку функций для целочисленного вещественного и строкового массива. В качестве исходных данных задается количество элементов массива и тип данных элементов массива.

#### 2) Курсовая работа «Структурное программирование на языке C++»

**Задание 1.** Разработайте и протестируйте программу, в которой над функцией в заданном интервале значений аргумента выполняются следующие операции:

1. Вывод значений аргумента и заданной функции  $F(x)$  с шагом  $h$ , начиная от начального значения  $a$  до конечного значения аргумента  $b$  в виде таблицы.
2. Вычисление корней уравнения  $F(x)=0$  методом половинного деления с точностью 0,0001.
3. Вывод на экран количества экстремумов функции  $F(x)$ .

4. Вычисление интеграла функции методом прямоугольников с точностью 0,001 на отрезке между вторым и третьим корнем функции.
5. Вычисление интеграла функции методом трапеций с точностью 0,001 на отрезке между вторым и третьим корнем функции.
6. Вывод графика функции  $F(x)$  на заданном интервале значений аргументов.
7. Вывод графика производной  $F'(x)$  функции  $F(x)$ .

Программа должна содержать меню, при выборе одного из пунктов которого, выполняется одна из операций над функцией  $F(x)$ .

### ВОПРОСЫ К ЭКЗАМЕНУ

1.	Операторы языка C++.
2.	Форматированный ввод/вывод данных. Поточный ввод/вывод данных.
3.	Массивы: объявление, индексирование, инициализация в C++
4.	Обработка одномерных и двумерных массивов
5.	Указатели: синтаксис и семантика типа, объявление и инициализация переменных в C++
6.	Операторы для работы с указателями в C++
7.	Строковый тип данных. Обработка строк.
8.	Структуры и объединения.
9.	Функции. Механизм передачи параметров.
10.	Перегрузка функций.
11.	Рекурсивные функции.
12.	Программирование доступа к файлам. Текстовые файлы.
13.	Программирование доступа к файлам. Бинарные файлы.
14.	Оценка сложности алгоритмов.
15.	Основы доказательства правильности алгоритмов.

### Модуль Объектно-ориентированное программирование

### ВОПРОСЫ К ЭКЗАМЕНУ

1.	Основные идеи ООП: использование объекта в качестве основной компоненты программы и децентрализация управления, реализуемая представлением программы совокупностью взаимодействующих объектов
2.	Сущность ООП: моделирование поведения объектов реального мира, наглядное представление объектов, создание и использование АТД, производные АТД
3.	Концепции ООП. Инкапсуляция и расширяемость типов
4.	Концепции ООП. Семантика объекта: данные как отражение состояния объекта и методы как средства обеспечения взаимодействия объектов. Представление данных. Классификация методов: конструкторы, деструкторы, селекторы, модификаторы
5.	Концепции ООП. Классы объектов: назначение и семантика, классы и абстрактные типы данных, наглядное представление классов. Объекты как экземпляры классов. Основные действия с объектами: создание, инициализация, обработка, разрушение
6.	Концепции ООП. Конструкторы и деструкторы, основное назначение
7.	Концепции ООП. Перегрузка операторов и функций. Ключевые слова: friend, operator
8.	Концепции ООП. Шаблоны и обобщенное программирование. Ключевое слово: template. Контейнерные классы
9.	Концепции ООП. Основные составляющие STL: контейнеры, итераторы, алгоритмы
10.	Концепции ООП. Наследование: базовый и производный класс, иерархия классов
11.	Концепции ООП. Полиморфизм. Абстрактный базовый класс

12.	Концепции ООП. Обработка исключительных ситуаций, проверка утверждений
13.	Концепции ООП. Преимущества и недостатки ООП. Основные свойства языка ООП. Место языка Си++ во внедрении и развитии средств ООП. Значение языка Си++ для профессиональной подготовки программистов
14.	Классы: компонентные данные, компонентные функции, способы задания доступа, синтаксическая структура определения класса, объявление, определение и инициализация объектов, объекты объявленные как константы
15.	Область видимости класса: оператор разрешения области видимости ( : : ) - унарная форма, бинарная форма; вложенные классы - классы, вложенные в классы, классы, вложенные в блоки, вложенные определения функций
16.	Компонентные данные: назначение, основные свойства, синтаксис и семантика определений компонентных данных, статические компонентные данные, статические константы, изменяемые компонентные данные (mutable)
17.	Компонентные функции: назначение, основные свойства, доступ к компонентным функциям, доступ компонентных функций к компонентным данным - функции селекторы, функции модификаторы
18.	Встроенные компонентные функции: назначение, синтаксис и семантика определения, рекомендации по использованию; компонентные функции, определяемые вне класса: назначение, синтаксис и семантика определения и объявления, рекомендации по использованию
19.	Указатель this, компонентные функции типа static и const (статические и постоянные компонентные функции): назначение, синтаксис и семантика определения и объявления, рекомендации по использованию
20.	Конструкторы: назначение, наличие нескольких конструкторов у одного класса - причины, разделение функций, способы задания конструктора: по умолчанию или явное
21.	Явное описание (определение или объявление) конструктора, размещение определения в теле класса и влияние размещения на функции конструктора, размещение определения вне своего класса, параметры конструктора и возможности их использования
22.	Конструкторы для инициализации объектов, инициализаторы конструктора; конструкторы для преобразования типов, ключевое слово explicit
23.	Конструкторы умолчания: назначение, способы задания, правила описания
24.	Конструкторы копирования: назначение, способы задания, правила описания
25.	Вызов конструктора: назначение, способы реализации, задание аргументов; явный и автоматический вызовы: ситуации использования, рекомендации по применению
26.	Деструкторы: назначение, основные свойства, способы задания: по умолчанию или явное
27.	Явное описание (определение или объявление) деструктора, размещение определения в теле класса и влияние размещения на функции деструктора, размещение определения вне своего класса, параметры деструктора и возможности их использования
28.	Вызов деструктора: назначение, способы реализации, явный и автоматический вызовы: ситуации использования, рекомендации по применению
29.	Специальный полиморфизм – преобразование АТД к встроенному типу
30.	Специальный полиморфизм – перегрузка функций, алгоритм выбора перегруженной функции
31.	Дружественные функции: основные свойства, номенклатура, необходимость, полезность и опасность применения, наборы дружественных функций, дружественные классы
32.	Дружественные функции: объявление – размещение, синтаксис, семантика,

	требования к параметрам и типу возвращаемого значения; определение – размещение, область видимости; вызов функции
33.	Специальный полиморфизм – перегрузка операторов: синтаксис, семантика, приоритет и порядок выполнения, доступные и недоступные для перегрузки операторы, перегружаемые как обычные функции, только как компонентные функции
34.	Перегрузка унарных и бинарных операторов: синтаксис и семантика
35.	Перегрузка операторов присваивания и индексирования: синтаксис и семантика
36.	Перегрузка операторов ввода-вывода (<<, >>): синтаксис и семантика
37.	Перегрузка оператора () для индексирования: синтаксис и семантика
38.	Перегрузка операторов доступа к элементу класса через указатель на объект →: синтаксис и семантика
39.	Перегрузка операторов new и delete: синтаксис и семантика
40.	Параметрический полиморфизм – шаблонные классы и шаблонные функции: назначение, параметризованные типы данных, синтаксис и семантика
41.	Шаблоны классов: назначение, синтаксис и семантика, объявление и определение шаблона, дружественные шаблонному классу функции, статические элементы шаблонного класса, аргументы шаблона класса
42.	Шаблоны функций: назначение, синтаксис и семантика, объявление, определение, вызов и перегрузка шаблонных функций, аргументы шаблона функции
43.	Наследование и иерархия классов: основные элементы методологии объектно-ориентированного проектирования, реализация механизма повторного использования кода, полиморфная обработка родственных объектов – чистый полиморфизм
44.	Отношение наследования: определение, семантика, базовый и производный классы, доступ к элементам классов, преимущества использования производных классов
45.	Построение производного класса специализацией базового, построение базового класса обобщением производного, иерархия классов
46.	Правила преобразования типов и видимость в иерархии родственных классов; конструкторы и деструкторы производного и базового классов: определение, вызов
47.	Наследование интерфейса, наследование реализации: назначение, особенности применения, достоинства и недостатки
48.	Виртуальные функции: назначение, определение, семантика, вызов; чисто виртуальные функции - отложенные методы
49.	Абстрактные базовые классы: назначение, определение, семантика, применение
50.	Шаблоны и наследование: достоинства и недостатки каждого механизма, формы комбинирования и особенности их применения
51.	Множественное наследование: причины появления и назначение, определение отношения множественного наследования: синтаксис и семантика
52.	Множественное наследование: виртуальное наследование, виртуальные функции при множественном наследовании, порядок выполнения конструкторов, практическое использование множественного наследования
53.	Конструирование объектно-ориентированных программ: структура ООП - разделение определений классов и определений компонентных функций; области видимости и время существования переменных - локальные, статические и глобальные переменные; организация файловой структуры программы для эффективной разработки, отладки и модификации
54.	Конструирование иерархий классов: принципы построения, рекомендации, примеры; создание и использование объектов во время исполнения программы: средства программирования и рекомендации по их использованию, организация массивов объектов; использование конструкторов, вызывающих другие конструкторы
55.	Идентификация типа на этапе выполнения: назначение, операторы - dynamic_cast,

	typeid, класс type_info
56.	Основные особенности использования функций, связанные с наследованием
57.	Обработка исключений: проверка утверждений (assert.h), сигналы - асинхронные исключения (signal.h)
58.	Механизм обработки исключений: номенклатура ситуаций и модели управления ими, средства поддержки управления в языке Си++
59.	Запуск исключений, перезапуск исключений, выражения исключений: синтаксис и семантика, спецификация исключения
60.	Пробные блоки try, обработчики исключений catch: синтаксис и семантика, иерархии классов для управления исключительными ситуациями
61.	Обработчики terminate() и unexpected(): назначение, особенности применения
62.	Использование стандартных исключений: отсутствие свободной памяти, тестирование приведений типов, логические ошибки, ошибки времени выполнения
63.	Обработка исключений как механизм восстановления после сбоев и как механизм передачи управления, модель завершения, используемая в C++
64.	Итераторы: назначение, синтаксис и семантика, объявления, определение, вызов, особенности применения
65.	Стандартная библиотека шаблонов: последовательные и ассоциативные контейнеры, интерфейсы контейнеров, адаптеры контейнеров
66.	Стандартная библиотека шаблонов: итераторы ввода, итераторы вывода, итераторы прохода вперед, двусторонние итераторы, итераторы произвольного доступа, итераторы istream_iterator и ostream_iterator, адаптеры итераторов
67.	Стандартная библиотека шаблонов: алгоритмы сортировки, алгоритмы, не изменяющие последовательность содержимого контейнера, алгоритмы, изменяющие последовательность содержимого контейнеров, численные алгоритмы
68.	Стандартная библиотека шаблонов: арифметические, сравнивающие, логические объекты-функции, адаптеры объектов-функций
69.	Объектно-ориентированное проектирование: CRC-карточки, диаграммы Вассермана-Пирчера, унифицированный язык моделирования (UML)
70.	Принципы и штампы объектно-ориентированного проектирования, тенденции развития методов, языков и инструментальных средств ООП, автоматизация разработки программ, визуализация ООП

## **КУРСОВАЯ РАБОТА**

Выполнение курсовой работы имеет своей целью систематизацию, закрепление и расширение теоретических знаний, получение практических навыков использования инструментальных средств вычислительной техники для решения прикладных задач, подбор и самостоятельное изучение технической литературы.

## **ЗАДАНИЯ**

**на курсовую работу по дисциплине «Программирование»  
Модуль: «Объектно-ориентированное программирование»**

**Тема: «Объектно-ориентированное программирование на языке Си ++»**

Общие требования к программному обеспечению для разработки **технического задания 1:**

- A. Разработайте иерархию родственных типов, корневой класс которой - абстрактный базовый класс, для моделирования и обработки данных предметной области согласно варианта A.x.x
- B. Создайте обобщенный контейнерный класс (базовый) и от него, используя закрытое наследование, производный класс – шаблон для хранения указателей на абстрактный базовый класс согласно варианта B.x.
- C. Для хранения объектов каждого производного класса используйте структуру данных согласно варианта C.x.
- D. Реализуйте функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
- E. Реализуйте файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
- F. Предусмотрите обработку различных исключительных ситуаций.
- G. Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

**Примечание:**

В части A – для создания иерархии родственных типов могут использоваться механизмы как одиночного, так и множественного наследования, рекомендуемое количество уровней иерархии: 3-4.

В части B – число элементов контейнерного класса определяется количеством производных классов иерархии, объекты которых будут создаваться для хранения данных предметной области.

В части C – количество объектов производных классов задается необходимым для тестирования функций обработки данных.

В части D – необходимо предусмотреть полиморфную обработку родственных объектов.

Общие требования к программному обеспечению для разработки **технического задания 2:**

- A. Разработайте иерархию родственных типов, корневой класс которой абстрактный базовый класс (**класс-интерфейс**), для моделирования и обработки данных предметной области набором отложенных методов - полиморфная обработка родственных объектов (согласно варианта A.x.x).
- B. Создайте обобщенный (**void\***) контейнерный класс (базовый) и от него, используя закрытое наследование, производный класс – шаблон для хранения указателей на абстрактный базовый класс-интерфейс (согласно варианта B.x).
- C. Для хранения объектов каждого производного класса используйте структуру данных (согласно варианта C.x).
- D. Реализуйте функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
- E. Реализуйте файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
- F. Предусмотрите обработку различных исключительных ситуаций.
- G. Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

**Примечание:**

В части A – для создания иерархии родственных типов могут использоваться механизмы как одиночного, так и множественного наследования, рекомендуемое количество уровней иерархии: 4-5.

В части В – число элементов контейнерного класса определяется количеством производных классов иерархии, объекты которых будут создаваться для хранения данных предметной области.

В части С – количество объектов производных классов задается необходимым для тестирования функций обработки данных.

#### Общие требования к программному обеспечению для разработки **технического задания 3:**

- A. Разработайте иерархии родственных типов, корневой класс которых **класс-интерфейс**, для моделирования и обработки данных предметных областей набором отложенных методов - полиморфная обработка родственных объектов (согласно вариантов **A.x.x** и **A.y.y**). Организуйте взаимодействие иерархий типов через указатели на интерфейсы и виртуальные функции (отложенные методы).
- B. Создайте обобщенный (**void\***) контейнерный класс (базовый) и от него, используя закрытое наследование, производный класс – шаблон для хранения указателей на абстрактный базовый класс - интерфейс (согласно варианта **B.x**).
- C. Для хранения объектов каждого производного класса используйте структуру данных (согласно варианта **C.x**).
- D. Реализуйте функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
- E. Реализуйте файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
- F. Предусмотрите обработку различных исключительных ситуаций.
- G. Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

#### **Примечание:**

В части А – для создания иерархии родственных типов могут использоваться механизмы как одиночного, так и множественного наследования, рекомендуемое количество уровней иерархии: **4-5**.

В части В – число элементов контейнерного класса определяется количеством производных классов иерархии, объекты которых будут создаваться для хранения данных предметной области.

В части С – количество объектов производных классов задается необходимым для тестирования функций обработки данных.

#### Общие требования к программному обеспечению для разработки **технического задания 4:**

- A. Разработайте иерархию родственных типов, корневой класс которой абстрактный базовый класс (**класс-интерфейс**), для моделирования и обработки данных предметной области набором отложенных методов - полиморфная обработка родственных объектов (согласно варианта **A.x.x**).
- B. Для хранения объектов каждого производного класса используйте структуру данных (согласно варианта **C.x**).
- C. Реализуйте функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
- D. Реализуйте файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
- E. Предусмотрите обработку различных исключительных ситуаций.
- F. Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

**Примечание:**

В части А – для создания иерархии родственных типов могут использоваться механизмы как одиночного, так и множественного наследования, рекомендуемое количество уровней иерархии: **4-5**.

В части В – количество объектов производных классов задается необходимым для тестирования функций обработки данных.

Общие требования к программному обеспечению для разработки **технического задания 5:**

- А. Разработайте иерархию родственных типов, корневой класс которой абстрактный базовый класс (**класс-интерфейс**), для моделирования и обработки данных предметной области набором отложенных методов - полиморфная обработка родственных объектов (согласно варианту **А.х.х**).
- В. Реализуйте функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
- С. Реализуйте файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
- Д. Предусмотрите обработку различных исключительных ситуаций.
- Е. Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

**Примечание:**

В части А – для создания иерархии родственных типов могут использоваться механизмы как одиночного, так и множественного наследования, рекомендуемое количество уровней иерархии: **4-5**.

Общие требования к программному обеспечению для разработки **технического задания 6:**

- А. Разработайте иерархию родственных типов для моделирования и обработки данных предметной области согласно варианту **А.х.х**.
- В. Для хранения объектов каждого производного класса используйте структуру данных согласно варианту **С.х**.
- С. Реализуйте функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
- Д. Реализуйте файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
- Е. Предусмотрите обработку различных исключительных ситуаций.
- Ф. Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

**Примечание:**

В части А – для создания иерархии родственных типов могут использоваться механизмы как одиночного, так и множественного наследования, рекомендуемое количество уровней иерархии: **3-4**.

В части В – количество объектов производных классов задается необходимым для тестирования функций обработки данных.

Общие требования к программному обеспечению для разработки **технического задания 7:**

- А. Разработайте иерархию родственных типов для моделирования и обработки данных предметной области согласно варианту **А.х.х**.



- В. Реализуйте функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
- С. Реализуйте файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
- Д. Предусмотрите обработку различных исключительных ситуаций.
- Е. Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

**Примечание:**

В части А – для создания иерархии родственных типов могут использоваться механизмы как одиночного, так и множественного наследования, рекомендуемое количество уровней иерархии: **3-4**.

Количество объектов производных классов задается необходимым для тестирования функций обработки данных.

Оформление пояснительной записки должно выполняться в соответствии с Указаниями к выполнению курсового проекта (работы).

### **Варианты для уточнения требований**

- А 1. Аппаратное обеспечение вычислительной техники
- А 2. Программное обеспечение вычислительной техники
- А 3. Элементная база для сборки компьютеров
- А 4. Инструментальные среды для разработки программного обеспечения
- А 5. Видеомониторы
- А 6. Печатающие устройства
- А 7. Принтеры
- А 8. Плоттеры
- А 9. Запоминающие устройства
- А 10. Устройства для управления компьютером
- А 11. Устройства передачи данных
- А 12. Аппаратное обеспечение компьютерных сетей
- А 13. Звуковоспроизводящая аппаратура
- А 14. Звукозаписывающая аппаратура
- А 15. Телефонные аппараты
- А 16. Осветительные приборы
- А 17. Оптические приборы
- А 18. Нагревательные приборы
- А 19. Холодильная техника
- А 20. Бытовая техника
- А 21. Вычислительные машины
- А 22. Электрические машины
- А 23. Подъемно-транспортные машины
- А 24. Строительные машины
- А 25. Металлургические машины
- А 26. Сельскохозяйственные машины
- А 27. Железнодорожный транспорт
- А 28. Автомобильный транспорт
- А 29. Воздушный транспорт
- А 30. Водный транспорт

- А 31. Энергетические установки
- А 32. Элементы интерьера
- А 33. Офисная мебель
- А 34. Электроинструмент
- А 35. Канцелярские товары
- А 36. Изделия целлюлозно-бумажной промышленности
- А 37. Строительные материалы
- А 38. Строительные конструкции
- А 39. Металлургические технологии
- А 40. Врачебный контроль физиологического состояния спортсмена
- А 41. Печи
- А 42. Каминь

- В 1. Статический вектор
- В 2. Динамический вектор
- В 3. Линейный односвязный список
- В 4. Линейный двусвязный список
- В 5. Стек
- В 6. Очередь

- С 1. Статический вектор
- С 2. Динамический вектор
- С 3. Статическая матрица
- С 4. Динамическая матрица
- С 5. Линейный односвязный список
- С 6. Линейный двусвязный список
- С 7. Стек
- С 8. Очередь
- С 9. Закольцованный список
- С 10. Двоичное дерево

Курсовая работа выполняется и оформляется в соответствии с Указаниями к выполнению курсового проекта (работы), защита включает в себя проверку отчета, тестирование программного кода и опрос по теории.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ ДЛЯ ТЕСТИРОВАНИЯ (ПРИМЕРЫ)**

1. Какие виды связей могут быть между классами?
2. В каком случае возникает иерархия родственных классов?
3. Что общего у класса-интерфейса и абстрактного базового класса?
4. Как увеличить зависимость классов друг от друга, не используя передачу сообщений?
5. Каким образом объект одного класса может передать сообщение объекту другого класса?  
и др.

### **3.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

## 1) Отчет по лабораторной работе от 0 до 5 баллов за 1 лабор.работу

Количество баллов	Показатели
5	Студент знает и формулирует основные методы и алгоритмы для решения задач по заданной теме, умеет анализировать методы решения задач и выбирать наиболее оптимальный метод для решения конкретной задачи. Анализирует полученные в ходе выполнения лабораторной работы, результаты, делает выводы.
4	Студент знает и формулирует основные методы и алгоритмы для решения задач по заданной теме, испытывает трудности при анализе методов решения задач и выборе наиболее оптимального метода для решения конкретной задачи. Анализирует полученные в ходе выполнения лабораторной работы, результаты, делает выводы.
3	Студент знает и формулирует основные методы и алгоритмы для решения задач по заданной теме, не умеет анализировать методы решения задач и выбирать наиболее оптимальный метод для решения конкретной задачи.
2	Студент знает и формулирует основные методы и алгоритмы для решения задач по заданной теме, не умеет анализировать методы решения задач и выбирать наиболее оптимальный метод для решения конкретной задачи.
1	Студент знает некоторые методы и алгоритмы для решения задач по заданной теме, не умеет анализировать методы решения задач и выбирать наиболее оптимальный метод для решения конкретной задачи.
0	Студент не знает методов и алгоритмов для решения задач по заданной теме, не умеет анализировать методы решения задач и выбирать наиболее оптимальный метод для

	решения конкретной задачи.
--	----------------------------

## 2) Зачет

40 – 35 баллов выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.

34 – 28 баллов выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.

27 – 20 баллов выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.

19 – 0 баллов выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, не выполняет практические работы.

## 3) Курсовая работа от 0 до 50 баллов

Показатели	Количество баллов
Постановка задачи, описание предметной области	0 - 5
Выбор метода решения задачи	0 - 5
Разработка алгоритма	0 - 15
Реализация алгоритма на языке программирования	0 - 15
Составление документации к разработанному программному обеспечению	0 - 10

#### 4) Экзамен

40 – 35 баллов (оценка «отлично») выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.

34 – 28 баллов (Оценка «хорошо») выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.

27 – 20 баллов (Оценка «удовлетворительно») выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической

последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.

19 – 0 баллов (Оценка «неудовлетворительно») выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, не выполняет практические работы.

### 3.4 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Показатели и критерии оценивания компетенций

ОПК-3 Готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

Показатели сформированности компетенций	Критерии оценивания компетенций		
	Недостаточный уровень	Базовый уровень	Повышенный уровень
<b>Умение</b> использовать технологии структурного, модульного и объектно-ориентированного программирования для создания кода, тестировать, документировать и получить решение поставленных задач различной сложности и объема;	Не умеет использовать технологии структурного, модульного и объектно-ориентированного программирования для создания кода, тестировать, документировать и получить решение поставленных задач;	Умеет использовать технологии структурного, модульного и объектно-ориентированного программирования для создания кода, тестировать, документировать и получить решение поставленных задач;	Умеет использовать технологии структурного, модульного и объектно-ориентированного программирования для создания кода, тестировать, документировать и получить решение поставленных задач различной сложности и объема;
<b>Владение</b> языками процедурного и объектно-ориентированного программирования;	Не владеет языками процедурного и объектно-ориентированного программирования;	Владеет языками процедурного и объектно-ориентированного программирования;	Обладает способностью широко использовать навыки владения языками процедурного и объектно-ориентированного программирования;

**ПК-1** Готовность применять основные методы и инструменты разработки программного обеспечения

Показатели сформированности компетенций	Критерии оценивания компетенций		
	Недостаточный уровень	Базовый уровень	Повышенный уровень
<b>Умение</b> применять основные методы и инструменты разработки программного обеспечения к решению практических задач.	Не умеет применять основные методы и инструменты разработки программного обеспечения к решению практических задач.	Умеет применять основные методы и инструменты разработки программного обеспечения к решению практических задач	Умеет применять основные методы и инструменты разработки программного обеспечения к решению практических задач и обосновывать выбор алгоритма для решения той или иной задачи

**ПК-12** Способность к формализации в своей предметной области с учетом ограничений используемых методов исследования

Показатели сформированности компетенций	Критерии оценивания компетенций		
	Недостаточный уровень	Базовый уровень	Повышенный уровень
<b>Владение навыками</b> формализованного представления информации и процесса обработки данных в виде программы на языке программирования	Испытывает затруднения при решении задач формализации представления информации и процессов обработки данных	Владеет навыками формализованного представления информации и процесса обработки данных в виде программы на языке программирования	Владеет навыками и успешно решает задачи формализованного представления информации и процесса обработки данных различного уровня сложности

**ПК-13** Готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности

Показатели сформированности компетенций	Критерии оценивания компетенций		
	Недостаточный уровень	Базовый уровень	Повышенный уровень
<b>Знание</b> на уровне представлений: современные технические и программные средства	Не знает современные технические и программные средства	Знает современные технические и программные средства взаимодействия с	Свободно оперирует знаниями современных технических и программных

Показатели сформированности компетенций	Критерии оценивания компетенций		
	Недостаточный уровень	Базовый уровень	Повышенный уровень
взаимодействия с компьютером, основные концепции структурного и объектно-ориентированного программирования;	взаимодействия с компьютером, основные концепции структурного и объектно-ориентированного программирования;	компьютером, основные концепции структурного и объектно-ориентированного программирования;	средств взаимодействия с компьютером, основных концепций структурного и объектно-ориентированного программирования;
<b>Знание</b> на уровне воспроизведения: современные технологии разработки алгоритмов и программ, методы тестирования, отладки и испытания программ,	Не знает современные технологии разработки алгоритмов и программ, методы тестирования, отладки и испытания программ,	Знает современные технологии разработки алгоритмов и программ, методы тестирования, отладки и испытания программ,	Знает, анализирует и объясняет выбор современных технологий разработки алгоритмов и программ, методов тестирования, отладки и испытания конкретных программ,
<b>Знание</b> на уровне понимания: технологии структурного, модульного и объектно-ориентированного программирования;	Не знает технологии структурного, модульного и объектно-ориентированного программирования;	Знает технологии структурного, модульного и объектно-ориентированного программирования;	Знает, понимает, объясняет технологии структурного, модульного и объектно-ориентированного программирования;
<b>Умение</b> использовать технологии структурного, модульного и объектно-ориентированного программирования для решения практических задач.	Не умеет использовать технологии структурного, модульного и объектно-ориентированного программирования для решения практических задач.	Умеет использовать технологии структурного, модульного и объектно-ориентированного программирования для решения практических задач.	Умеет выбирать и использовать технологии структурного, модульного и объектно-ориентированного программирования для решения практических задач, обосновывать свой выбор.

**ПК-14** Готовность обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности

Показатели	Критерии оценивания компетенций
------------	---------------------------------



<b>сформированности компетенций</b>	<b>Недостаточный уровень</b>	<b>Базовый уровень</b>	<b>Повышенный уровень</b>
<b>Умение</b> обосновать выбор алгоритма решения прикладной задачи;	Не умеет обосновывать выбор алгоритма решения прикладной задачи;	Умеет обосновывать выбор алгоритма решения прикладной задачи;	Свободно оперирует знаниями при выборе и обосновании выбора алгоритма решения прикладной задачи
<b>Владение</b> навыками самостоятельной разработки алгоритма решения прикладной задачи;	Не владеет навыками самостоятельной разработки алгоритма решения прикладной задачи;	Владеет навыками самостоятельной разработки алгоритма решения прикладной задачи;	Обладает способностью широко использовать навыки самостоятельной разработки алгоритма решения прикладных задач различного уровня сложности.

### Шкала оценивания компетенций:

Оценка в 100-балльной шкале	Оценка в 5-ти балльной шкале	Уровень сформированности компетенций
0-54 баллов	неудовлетворительно (не зачтено)	недостаточный
55-69 баллов	удовлетворительно (зачтено)	базовый
70-85 баллов	хорошо (зачтено)	повышенный
86-100 баллов	отлично (зачтено)	

#### **4 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Организация занятий по дисциплине носит циклический характер. По всем разделам предусмотрена взаимоувязанная цепочка учебных работ: лекция – самостоятельная работа студентов (аудиторная и внеаудиторная). Для своевременной помощи студентам при изучении дисциплины организуются индивидуальные и групповые консультации.

По итогам изучения всех модулей осуществляется аттестация студента в форме зачета или экзамена.

Так как дисциплина формирует профессиональные компетенции, к её изучению предъявляются следующие организационные требования:

- обязательное посещение студентом всех видов аудиторных занятий;
  - ведение конспекта в ходе лекционных занятий;
  - качественная самостоятельная подготовка к практическим и лабораторным занятиям, активная работа на них;
  - своевременная сдача преподавателю отчетных документов по аудиторным и внеаудиторным видам работ;
  - в случае наличия пропущенных студентом занятий, необходимо получить консультацию по подготовке и оформлению отдельных видов заданий.
- Для успешного освоения курса, студенту предлагаются учебная и учебно-методическая литература по всем разделам, электронные учебно-методические материалы на образовательном портале ЧГУ на сайте «Программирование».

## **5 УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**

### **5.1 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)**

Основная литература:

1. Павловская, Т.А. С/С++.Программирование на языке высокого уровня : учебник для вузов. - СПб. : Питер, 2007. - 461 с. : ил. + Приложения. - (Учебник для вузов). - Библиогр.: с. 383. - ISBN 5-94723-568-4. - ISBN 978-5-94723-568-5.
2. Хорев П.Б. Объектно-ориентированное программирование : учебное пособие для вузов. - 4-е изд. - Москва : ИЦ "Академия", 2012. - 447 с. - (Высшее профессиональное образование. Бакалавриат) (Информатика и вычислительная техника). - Библиогр.: с.444-445. - ISBN 978-5-7695-9265-2

Дополнительная литература:

1. Александров Э. Э. , Афонин В. В. Программирование на языке С в Microsoft Visual Studio 2010: учебное пособие. - М.: Интернет-Университет Информационных Технологий, 2010. – 500 с. [http://biblioclub.ru/index.php?page=book\\_red&id=233564](http://biblioclub.ru/index.php?page=book_red&id=233564)
2. Иванова, Г.С. Основы программирования : учебник для студентов вузов. - 2-е изд., перераб. и доп. - Москва : МГТУ им. Н.Э. Баумана, 2002. - 415 с. : ил. - (Информатика в техническом университете). - Библиогр. : с. 413. - ISBN 5-7038-1957-1.
3. Давыдов, В.Г. Программирование и основы алгоритмизации : учебное пособие для вузов / Давыдов В.Г. - 2-е изд. - Москва : Высшая школа, 2005. - 448 с. : ил. + Приложения. - Библиогр.: с.442. - ISBN 5-06-004432-7.

### **5.2 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)**

- 1 Е.В. Ершов, Л.Н. Виноградова и др. Методика и организация самостоятельной работы студентов. Учеб.-метод.пособие. Череповец: ЧГУ, 2012. – 208 с.
- 2 Е.В. Ершов, Л.Н. Виноградова. Указания к выполнению курсового проекта (работы): Учеб.-метод. пособие. Череповец: ЧГУ, 2004
- 3 Е.В. Ершов, Л.Н. Виноградова. Методические указания к выполнению лабораторных работ по дисциплине «Объектно-ориентированное программирование». Ч. 1: Учеб.-метод. пособие. Череповец: ЧГУ, 2003. с. 27.
- 4 Е.В. Ершов, О.Г. Ганичева, Л.Н. Виноградова. Методические указания к выполнению лабораторных работ по дисциплине «Объектно-ориентированное программирование. Лабораторный практикум. Часть 2» Учеб.-метод. пособие. Череповец: ГОУ ВПО ЧГУ, 2007. с. 27.

### **5.3 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)**

- 1 поисковые системы [www.google.ru](http://www.google.ru), [www.yandex.ru](http://www.yandex.ru).
- 2 Образовательный портал ЧГУ [edu.chsu.ru](http://edu.chsu.ru)
- 3 Электронный каталог библиотеки университета  
<http://marc.chsu.ru/Default.asp>
- 4 Научная электронная библиотека «eLibrary» <http://elibrary.ru/defaultx.asp>
- 5 Информационная система «Единое окно» <http://window.edu.ru/>
- 6 Электронная библиотека «Лань» (бесплатный пакет) <http://e.lanbook.com/>
- 7 Новая электронная библиотека в ЧГУ <http://biblioclub.ru/>

**6 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ), включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

- ОС семейства Microsoft Windows
- Microsoft Visual Studio 2012
- OpenOffice
- Google Chrome
- Far manager
- Eclipse IDE

## 7 ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Оборудованные учебные кабинеты, объекты для проведения практических занятий		Перечень основного оборудования
Наименование	Адрес	
Компьютерный класс	217	<p><b>I. Компьютер Некс премиум учебный</b> Intel Core i5-3330 3ГГц, 8Гб ОЗУ, 250Гб HDD Монитор жк 19" Philips 190V41 Количество: 15</p> <p><b>II. Компьютер Некс базовый рабочий</b> Intel Core i3-3210 3,2ГГц, 8Гб ОЗУ, 250Гб HDD Количество: 1 (системный блок)</p> <p><b>III. Компьютер</b> Intel Core 2 Duo 2,8ГГц, 2Гб ОЗУ, 150Гб HDD Монитор жк 19" Acer V193W Количество: 1</p> <p><b>IV. Комплекс «Активный экран» ACTIVboard 78</b> - сеть переменного тока от 100 до 240 вольт переменного тока - частота от 50 до 60 Гц - потребляемый ток 300 мА</p> <p><b>V. Проектор BENQ 622c DLP, XGA</b></p>
Компьютерный класс	218	<p><b>I. Компьютер «Базовый учебный»</b> Intel G2140 3,3 ГГц, 8Гб ОЗУ, 320 Гб HDD Монитор жк 19" AOC e941 va Количество: 14</p>
Компьютерный класс	218А	<p><b>I. Компьютер «Базовый учебный»</b> Intel G2140 3,3 ГГц, 8Гб ОЗУ, 320 Гб HDD Монитор жк 19" AOC e941 va Количество: 12</p>
Компьютерный класс	219	<p><b>I. Компьютер «Базовый учебный»</b> Intel G2140 3,3 ГГц, 8Гб ОЗУ, 320 Гб HDD Монитор жк 19" AOC e941 va Количество: 14</p> <p><b>II. Настенный экран</b> ScreenMedia Goldview, формат 183*244</p>