

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета информационных
технологий и компьютерной безопасности
 / П.Ю. Гусев /
И.О. Фамилия
«31» августа 2021 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

«Объектно-ориентированное программирование»

Направление подготовки 09.03.01 Информатика и вычислительная техника

Профиль Программное обеспечение автоматизированных систем

Квалификация выпускника бакалавр

Нормативный период обучения 4 года / 4 года и 11 м.

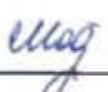
Форма обучения очная / заочная

Год начала подготовки 2019

Автор программы

 /Акинин А.А./

Заведующий кафедрой Ав-
томатизированных и вычис-
лительных систем

 /Подвальный С.Л./

Руководитель ОПОП

 /Гусев П. Ю./

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

формирование у обучающихся объектно-ориентированного мышления, изучение объектно-ориентированной методологии программирования, изучение ключевых понятий объектно-ориентированного программирования.

1.2. Задачи освоения дисциплины

– к теоретическим задачам относятся: ознакомление с тенденцией развития программного обеспечения; обучение разработке алгоритмов на основе объектно-ориентированного подхода; изучение основных технологий программирования C++, а также стандартов кодирования, спецификаций и последующих решений на их основе;

– прикладные задачи состоят в ознакомлении с современными программными средами, приобретении навыков проектирования, разработки и сопровождения программ в рамках объектно-ориентированного стиля программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Объектно-ориентированное программирование» относится к дисциплинам базовой части блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Объектно-ориентированное программирование» направлен на формирование следующих компетенций:

ОПК-8 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

ОПК-9 - Способен осваивать методики использования программных средств для решения практических задач.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-8	знать – язык программирования C++, операционную систему WINDOWS и оболочки, среду разработки программного обеспечения MS Visual Studio;
	уметь – составлять блок-схемы алгоритмов, разрабатывать алгоритмы с применением объектно-ориентированного подхода, писать и отлаживать коды на языке программирования C++, тестировать работоспособность программ в среде MS Visual Studio, использовать стандартные библиотеки языка C++;
	владеть

	– языком программирования C++; навыками отладки и тестирования программ в среде MS Visual Studio;
ОПК-9	знать – классификацию программных средств, порядок настройки и работы в интегрированных средах разработки программного обеспечения (на примере MS Visual Studio);
	уметь – находить в технической документации по использованию программного средства функции, необходимые для решения конкретной задачи;
	владеть – навыками разработки технической документации по использованию ПО для решения конкретной задачи; – навыками создания презентаций и видеороликов о методике использования программного средства для решения конкретной задачи.

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Объектно-ориентированное программирование» составляет 5 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		2
Аудиторные занятия (всего)	90	90
В том числе:		
Лекции	36	36
Лабораторные работы (ЛР)	54	54
Самостоятельная работа	54	54
Курсовой проект	+	+
Часы на контроль	36	36
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость:		
академические часы	180	180
зач.ед.	5	5

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		2
Аудиторные занятия (всего)	20	20
В том числе:		

Лекции	8	8
Лабораторные работы (ЛР)	12	12
Самостоятельная работа	151	151
Курсовой проект	+	+
Часы на контроль	9	9
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость:		
академические часы	180	180
зач.ед.	5	5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Интегрированные среды и средства разработки на языке программирования высокого уровня	Введение. Анализ языков программирования и обзор сред разработки программного обеспечения. Возможности и производительность современных компиляторов, сборщиков проектов. Среды разработки. Интерфейс и конфигурация интегрированной среды разработки (IDE) под решение вычислительных задач. Разработка консольных проектов и приложений с графическим интерфейсом	8	12	10	30
2	Базовые конструкции и ключевые слова языка программирования высокого уровня	Синтаксис и операции. Типы данных языка и расширения типов, предоставляемые средой. Арифметические операции. Правила именования методов, переменных и констант. Основные алгоритмические конструкции. Использование структур, условных конструкций, циклов, функций, а также директив препроцессора.	8	12	10	30
3	Объектно-ориентированный подход к разработке программных средств	Введение в классы и объекты. Описание и свойства объектов. Конструкторы, деструкторы классов, объявления статических объектов, переменных и методов в классе. Наследование в ООП. Наследование, виды наследования. Абстрактные классы. Множественное наследование. Управление данными, директивы препроцессора. Структуры, исключения, набор директив для сборки проектов.	8	10	12	30

4	Разработка приложений с графическими интерфейсными возможностями	Введение в разработку пользовательского интерфейса. Средства разработки интерфейса IDE сред. Понятие интерфейса пользователя, метафоры и поиск подобных решений, интерактивные средства разработки. Основы разработки программных интерфейсов, средства разработки, иерархия и структура интерфейсных классов. Обработка событий в приложениях. Виды приложений и особенности их создания. Средства взаимодействия между приложением и действиями, исходящими от пользователя. Однооконные и многооконные приложения. Обработка данных, запись результатов (SDI и MDI). Организация вывода графической информации. Представление, ввод и вывод данных на носители информации. Работа с графикой на плоскости, технологии реализации вывода в 3D, подключение к данным, создание архивов и извлечение информации из них, регулярные выражения проектам библиотек, поддерживающих OpenGL. Работа с файлами, поддержка баз.	6	10	10	26
5	Создание подсистем на основе подключаемых библиотек	Динамически подключаемые источники данных. Работа с системными библиотеками операционных систем.	6	10	12	28
Итого			36	54	54	144

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Интегрированные среды и средства разработки на языке программирования высокого уровня	Введение. Анализ языков программирования и обзор сред разработки программного обеспечения. Возможности и производительность современных компиляторов, сборщиков проектов. Среды разработки. Интерфейс и конфигурация интегрированной среды разработки (IDE) под решение вычислительных задач. Разработка консольных проектов и приложений с графическим интерфейсом	2	2	30	34
2	Базовые конструкции и ключевые слова языка программирования высокого уровня	Синтаксис и операции. Типы данных языка и расширения типов, предоставляемые средой. Арифметические операции. Правила именования методов, переменных и констант. Основные алгоритмические конструкции. Использование структур, условных конструкций, циклов, функций, а также директив препроцессора.	2	3	30	35
3	Объектно-ориентированный подход к разработке программных средств	Введение в классы и объекты. Описание и свойства объектов. Конструкторы, деструкторы классов, объявления статических объектов, переменных и методов в классе.	2	3	30	35

		Наследование в ООП. Наследование, виды наследования. Абстрактные классы. Множественное наследование. Управление данными, директивы препроцессора. Структуры, исключения, набор директив для сборки проектов.				
4	Разработка приложений с графическими интерфейсными возможностями	Введение в разработку пользовательского интерфейса. Средства разработки интерфейса IDE сред. Понятие интерфейса пользователя, метафоры и поиск подобных решений, интерактивные средства разработки. Основы разработки программных интерфейсов, средства разработки, иерархия и структура интерфейсных классов. Обработка событий в приложениях. Виды приложений и особенности их создания. Средства взаимодействия между приложением и действиями, исходящими от пользователя. Однооконные и многооконные приложения. Обработка данных, запись результатов (SDI и MDI). Организация вывода графической информации. Представление, ввод и вывод данных на носители информации. Работа с графикой на плоскости, технологии реализации вывода в 3D, подключение к данным, создание архивов и извлечение информации из них, регулярные выражения проектам библиотек, поддерживающих OpenGL. Работа с файлами, поддержка баз.	1	2	30	33
5	Создание подсистем на основе подключаемых библиотек	Динамически подключаемые источники данных. Работа с системными библиотеками операционных систем.	1	2	31	34
Итого			8	12	151	171

5.2 Перечень лабораторных работ

Лабораторная работа № 1. Знакомство со средой разработки Microsoft Visual Studio 2010. Создание консольных и графических приложений.

Лабораторная работа № 2. Изучение базовых инструкций языка C++. Особенности работы с массивами в языке C++.

Лабораторная работа № 3. Использование встроенных типов данных языка C++. Манипулирование основными системами счисления.

Лабораторная работа № 4. Символьные и строковые типы данных в языке C++. Основные функции работы со строками.

Лабораторная работа № 5. Работа с файлами в языке C++. Запись и чтение информации из файла.

Лабораторная работа № 6. Введение в понятие класса в языке C++.

Лабораторная работа № 7. Динамические структуры данных. Организация списка с помощью ООП.

Лабораторная работа № 8. Углубленное изучение концепций инкапсуляции, наследования и полиморфизма.

Лабораторная работа № 9. Изучение основных контейнеров и алгоритмов стандартной библиотеки шаблонов (STL).

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 2 семестре для очной формы обучения, в 2 семестре для заочной формы обучения.

Примерная тематика курсового проекта: «Применение объектно-ориентированной методологии при разработке программных модулей».

Задачи, решаемые при выполнении курсового проекта.

Разработать классы для описанных ниже объектов. Для всех объектов в классе должны быть реализованы следующие методы: конструктор по умолчанию, конструктор с параметрами, конструктор копирования, деструктор и методы `set (...)`, `get (...)`, `show (...)`. На основе разработанного класса необходимо создать массив экземпляров класса с возможностью добавления, удаления и редактирования элементов этого массива, а также дополнительно реализовать описанный функционал для каждого типа объекта.

1. Класс Студенты.

Свойства: фамилия, имя, отчество, дата рождения, адрес, телефон, факультет, курс, отметка иногородний или нет.

Реализовать:

- а) вывод списка студентов запрошенного факультета;
- б) вывод списка студентов запрошенного факультета и курса;
- в) вывод списка иногородних студентов.

2. Класс Абитуриенты.

Свойства: фамилия, имя, отчество, адрес, предметы, оценки.

Реализовать:

- а) вывод списка абитуриентов, имеющих неудовлетворительные оценки;
- б) вывод списка абитуриентов, сумма баллов у которых не меньше заданной;
- в) выбрать абитуриента, имеющего самую высокую сумму баллов.

3. **Класс Самолет.**

Свойства: пункт отправления, пункт назначения, номер рейса, количество пассажиров, день недели вылета, день недели прилета.

Реализовать:

- а) вывод списка рейсов для заданного пункта назначения;
- б) вывод списка рейсов для заданного дня недели;
- в) вывод списка рейсов которые начинаются в один день недели, а заканчиваются в другой.

4. **Класс Книга.**

Свойства: автор, название, издательство, год выпуска, количество страниц.

Реализовать:

- а) вывод список книг заданного автора;
- б) вывод списка книг, с количеством страниц меньше заданного;
- в) список книг, выпущенных после заданного года.

5. **Класс Сотрудник.**

Свойства: фамилия, имя, отчество, должность, подразделение, год поступления на работу, зарплата.

Реализовать:

- а) вывод списка сотрудников, работающих в заданном подразделении;
- б) вывод списка сотрудников, зарплата которых больше заданной;
- в) вывод списка сотрудников, занимающих заданную должность.

6. **Класс Вагон.**

Свойства: тип вагона, номер вагона, число мест в вагоне, фамилия проводника, номер поезда, к которому прикреплен вагон, признак наличия кондиционера, срок эксплуатации вагона.

Реализовать:

- а) вывод списка вагонов, находящихся в эксплуатации больше указанного количества лет;
- б) вывод списка вагонов, закрепленных за проводником с заданной фамилией;
- в) вывод списка вагонов с кондиционером.

7. **Класс Поезд.**

Свойства: пункт назначения, пункт отправления, номер поезда, протяженность маршрута, число общих мест, число купейных мест, число плацкартных мест.

Реализовать:

- а) вывод списка поездов, следующих до заданного пункта назначения;
- б) вывод общего числа мест в поезде с заданным номером;

в) вывод списка поездов, с протяженностью маршрута больше заданной

8. Класс Товар.

Свойства: наименование, производитель, цена, вес, срок хранения, количество. **Реализовать:**

- а) вывод списка товаров для заданного наименования;
- б) вывод списка товаров для заданного наименования, цена которых не превышает указанной;
- в) вывод списка товаров, срок хранения которых больше заданного.

9. Класс Пациент.

Свойства: фамилия, имя, отчество, адрес, номер медицинской карты, диагноз, фамилия лечащего врача.

Реализовать:

- а) вывод списка пациентов, имеющих данный диагноз;
- б) вывод информации о пациенте, по номеру номер медицинской карты;
- в) вывод списка пациентов заданного лечащего врача.

10. Класс Автобус.

Свойства: фамилия водителя, номер автобуса, номер маршрута, марка, год начала эксплуатации, пробег.

Реализовать:

- а) вывод списка автобусов для заданного номера маршрута;
- б) вывод списка автобусов, которые эксплуатируются больше заданного количества лет;
- в) вывод списка автобусов, пробег у которых больше заданного количества км.

11. Класс Заказ в ресторане.

Свойства: фамилия официанта, номер столика, наименование блюда, стоимость блюда, размер чаевых, признак оплаты наличными/по карте.

Реализовать:

- а) вывод информации о заказах, принятых конкретным официантом;
- б) вывод списка заказов, стоимость которых больше указанной;
- в) вывод списка клиентов, платящих наличными.

12. Класс Заказ такси.

Свойства: фамилия водителя, пункт отправления, пункт прибытия, номер машины, стоимость поездки, протяженность маршрута.

Реализовать:

- а) вывод информации о заказах, по номеру машины;
- б) вывод списка заказов, протяженность которых больше указанной;
- в) вывод размера среднего чека по всем заказам.

13. Класс Клиент банка.

Свойства: фамилия, имя, отчество, адрес, телефон, номер банковского счета, количество денег на счете, лимит овердрафта

Реализовать:

- а) вывод информации о клиенте по номеру банковского счета;
- б) вывод списка клиентов, количество денег на счете которых, меньше указанного;
- в) вывод списка клиентов, без овердрафта

14. Класс Файл.

Свойства: имя файла, расширение файла, размер файла, дата создания, количество обращений к файлу, признак возможности редактирования файла.

Реализовать:

- а) вывод список файлов, для которых запрещено редактирование;
- б) вывод списка файлов, размер которых превышает заданный;
- в) вывод списка файлов, число обращений к которым превышает заданное.

15. Класс Страница.

Свойства: количество букв, количество гласных, размер шрифта, число абзацев, номер страницы, число строк.

Реализовать:

- а) вывод списка страниц, содержащих количество букв больше заданного;
- б) вывод списка номеров страниц и количеств согласных букв на них;
- в) вывод среднего количества букв на странице.

16. Класс Квартира.

Свойства: номер квартиры, номер этажа, количество комнат, общая площадь, жилая площадь, фамилия владельца.

Реализовать:

- а) вывод списка квартир, имеющих заданное число комнат;
- б) вывод списка квартир, принадлежащих заданному владельцу;
- в) вывод средней площади комнаты квартиры с указанным номером.

17. Класс Телефон.

Свойства: производитель, модель, объем памяти, цвет корпуса, диагональ экрана, емкость аккумулятора.

Реализовать:

- а) вывод списка телефонов с заданным цветом корпуса;
- б) вывод списка телефонов заданного производителя с диагональю больше заданной;
- в) вывод списка телефонов с емкостью аккумулятора больше заданной.

18. **Класс Человек.**

Свойства: фамилия, имя, отчество, адрес, пол, образование, возраст.

Реализовать:

- а) вывод списка граждан, возраст которых превышает заданный;
- б) вывод списка граждан с высшим образованием;
- в) вывод списка граждан заданного пола.

19. **Класс Собака.**

Свойства: кличка, порода, вес, фамилия хозяина, окрас, пол, рост, возраст.

Реализовать:

- а) вывод списка собак, принадлежащих указанному владельцу;
- б) вывод списка собак, указанного окраса;
- в) вывод информации по собакам, вес которых больше заданного.

20. **Класс Дерево.**

Свойства: порода, высота, обхват ствола, отметка о выбраковке, возраст.

Реализовать:

- а) вывод списка деревьев, заданной породы, подлежащих спиливанию;
- б) вывод списка деревьев, возраст которых больше заданного;
- в) вывод информации по деревьям, обхват ствола которых больше заданного.

21. **Класс Двигатель.**

Свойства: производитель, мощность, объем цилиндров, тип двигателя(бензин/дизель), количество клапанов, количество цилиндров.

Реализовать:

- а) вывод списка двигателей, мощность которых больше указанной;
- б) вывод списка двигателей, с заданным количеством клапанов и цилиндров;
- в) вывод информации по двигателям, заданного типа.

22. **Класс Ракета.**

Свойства: производитель, мощность двигателя, дальность, гарантированная точность (в метрах), количество боеголовок, мощность боеголовок.

Реализовать:

- а) вывод списка ракет, с заданным количеством боеголовок и их количеством;
- б) вывод списка ракет, указанного производителя;
- в) вывод информации по ракетам, с дальностью больше указанной.

Курсовой проект включают в себя графическую часть и расчетно-пояснительную записку.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-8	знать – язык программирования C++, операционную систему WINDOWS и оболочки, среду разработки программного обеспечения MS Visual Studio;	Тестирование Проверка этапов выполнения курсового проекта Результаты проведения коллоквиума Ответы на теоретические вопросы при защите лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь – составлять блок-схемы алгоритмов, разрабатывать алгоритмы с применением объектно-ориентированного подхода, писать и отлаживать коды на языке программирования C++, тестировать работоспособность программ в среде MS Visual Studio, использовать стандартные библиотеки языка C++;	Эффективность использования изученного теоретического материала при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть – языком программирования C++; навыками отладки и тестирования программ в среде MS Visual Studio;	Владение языком программирования C++ при выполнении лабораторных работ Владение навыками отладки и тестирования работоспособности программы, реализованной на языке программирования C++, при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-9	знать – классификацию программных средств, порядок настройки и работы в интегрированных средах разработки программного обеспечения (на примере MS Visual Studio);	Тестирование Проверка этапов выполнения курсового проекта Результаты проведения коллоквиума	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

		Ответы на теоретические вопросы при защите лабораторных работ		
	уметь – находить в технической документации по использованию программного средства функции, необходимые для решения конкретной задачи;	Эффективность использования изученного теоретического материала при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть – навыками разработки технической документации по использованию ПО для решения конкретной задачи; – навыками создания презентаций и видеороликов о методике использования программного средства для решения конкретной задачи.	Владение способами описания методики использования программного средства при выполнении лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются во 2 семестре для очной формы обучения, 2 семестре для заочной формы обучения по четырехбалльной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ОПК-8	знать – язык программирования C++, операционную систему WINDOWS и оболочки, среду разработки программного обеспечения MS Visual Studio;	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь – составлять блок-схемы алгоритмов, разрабатывать алгоритмы с применением объектно-ориентированного подхода, писать и отлаживать коды на языке программирования C++, тестировать работоспособность программ в среде MS Visual Studio, использовать стандартные библиотеки языка C++;	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть – языком программирования C++; навыками отладки и тестирования	Решение прикладных задач в конкретной	Задачи решены в полном объеме и получены	Продемонстрирован верный ход решения всех,	Продемонстрирован вер-	Задачи не решены

	программ в среде MS Visual Studio;	предметной области	верные ответы	но не получен верный ответ во всех задачах	ный ход решения в большинстве задач	
ОПК-9	знать – классификацию программных средств, порядок настройки и работы в интегрированных средах разработки программного обеспечения (на примере MS Visual Studio);	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь – находить в технической документации по использованию программного средства функции, необходимые для решения конкретной задачи;	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть – навыками разработки технической документации по использованию ПО для решения конкретной задачи; – навыками создания презентаций и видеороликов о методике использования программного средства для решения конкретной задачи.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Выберите неверное высказывание?

- A. К элементам массива можно получить доступ с помощью арифметики указателей.
- B. Имя массива дает адрес первого элемента массива.
- C. Массив и указатель ничем не различаются. +
- D. Если массив передается в параметре функции, то он всегда передается как указатель

2. Правильно ли, что «Структура является собранием одной или нескольких переменных».

- A. Да +
- B. Нет
- C. Только нескольких

3. Верно ли, что «Для определения структуры нужно указать только данные для хранения переменных».

- A. Да+
- B. Нет
- C. Данные и их адрес

4. Что, помимо полей, могут включать в себя структуры?

- A. Указатель на ту же структуру +
- B. Заголовки функций +
- C. Еще одну структуру+
- D. Ничего

5. Чтобы обратиться к искомому элементу входящего в структуру массива нужно:

- A. Указать адрес массива
- B. Указать индекс
- C. Сделать все выше перечисленное +

6. Чем различаются операторы выбора «. » и «→» ?

- A. Оператор «→» применяется к объекту структурного типа, а оператор «. » к указателю на объект структурного типа.
- B. Операторы «. » и «→» применяются как к объекту структурного типа, так и к указателю на объект структурного типа.
- C. Оператор «. » применяется к объекту структурного типа, а оператор «→», к указателю на объект структурного типа.+

7. В программе описано класс и объект `class A {public: int a, b, c; }; A * obj;`
Как обратиться к атрибуту c?

- A. `obj.c`
- B. `obj→ c +`
- C. `obj A → → c`
- D. `obj→ A.c`

8. Свойство формы `name` – это...

- A. Имя формы, используется для управления формой и доступа к компонентам формы +
- B. Текст заголовка (надпись на форме)
- C. Значок в заголовке диалогового окна, обозначающий кнопку вывода системного меню

9. Какой оператор используется для динамического выделения массива объектов типа T?

- A. allocate T
- B. new T
- C. new T[x] +

10. Какое значение будет иметь p после выполнения следующего кода

```
int *p = 1000;  
p++;
```

- A. 0
- B. 1004 +
- D. 1001

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Какова причина появления свободных участков памяти между соседними полями в структуре.

- A. Из-за наличия требований выравнивания определенных типов; +
- B. Если поле структуры является массивом;
- C. Из-за не соблюдения правил описания полей структуры.

2. Для хранения значения переменной объединения выделяется памяти...

- A. Столько, чтобы вместить самый "широкий" элемент объединения; +
- B. Сумма значений занимаемых каждым элементом в отдельности;
- C. Столько, чтобы вместить первый элемент объединения.

3. С помощью чего реализуется принцип полиморфизма в C ++?

- A. Наличие множественного наследования.
- B. Наличие виртуальных методов. +
- C. Использование виртуального наследования.
- D. Наличие абстрактных классов.

4. Компонент, предоставляющий возможность ввода данных в поле редактирования путем набора на клавиатуре или выбором из списка

- A. Button
- B. Memo
- C. CheckBox
- B. ComboBox +

5. Свойство, определяющее высоту объекта в пикселах.

- A. Height +
- B. Width
- C. Color

6. Элемент управления: `timage`
A. Отображает графическое изображение на форме +
B. Служит для отображения простейших графических объектов на форме: окружность, квадрат и т.п.
C. Позволяет создать на форме прокручиваемую область с размерами большими, чем экран, на которой можно разместить объекты
7. Свойство компонента `TEdit`, указывающее цвет фона - ...
A. `Color+`
B. `Text`
E. `Height`
8. Компоненты, которые видны во время работы приложения, с ними напрямую может взаимодействовать пользователь, называются...
A. Визуальными +
B. Невизуальными
C. Активными
9. Выберите верное утверждение о деструкторе класса в `C++`.
A. Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.
B. Деструктор принимает в качестве параметра указатель `this`.
C. Деструктор не содержит параметров. +
D. Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.
10. Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренним компонентам?
A. Шаблонная.
B. Полиморфная.
C. Дружественная. +
D. Статическая.

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Какая из перечисленных функций не может быть конструктором?
A. `void String ()` +
B. `String ();`
C. `String (String & s)`
D. `String (const int a)`
2. Если в программе на языке `C++` в производном классе переопределена операция `new` то ...

- A. Все объекты этого класса и все объекты классов, выведенных из него, будут использовать эту операцию независимо от зоны видимости, в которой она переопределена.
- B. Производные от этого класса могут использовать глобальную операцию применив операцию базовый_класс :: new. +
- C. Операцию new нельзя переопределить.
- D. В любом случае эта операция будет доступна только в пределах класса-потомка.

3. Какой из перечисленных методов может быть конструктором для класса String в языке C ++?

- A. String * String ();
- B. void String ();
- C. String (String & s); +
- D. const String (int a);

4. Вызовет данный код ошибку компиляции? class Rectangle { public: int a, b; int sum (); int square (); ~ Rect (); };

- A. Ошибки нет, все записано верно.
- B. Ошибка является: имя деструктора должно совпадать с именем класса. +
- C. Ошибка является: имя деструктора не может начинаться с маленькой буквы.
- D. Ошибка является: никакой идентификатор в C ++ не может начинаться со знака «~».

5. Укажите правильное объявление виртуального метода, который принимает одно целочисленное значение и возвращает void.

- A. virtual void SomeFunction (int x); +
- B. void SomeFunction (int x) virtual;
- C. virtual SomeFunction (int x);
- D. virtual void SomeFunction (int * x);

6. Укажите правильное использование оператора friend.

- A. class A {int_friend CountPass (); private: short i;};
- B. class A {public: friend int H :: CountPass (); private: short i;}; +
- C. class A {public: int A1 :: CountPass (); friend: short i;};
- D. class A {public: friend int H :: q; short i;};

7. Принцип объектно-ориентированного программирования, заключающийся в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется:

- A. Наследование.
- B. Сочетание.

- C. Инициализация.
- D. Инкапсуляция. +

8. В программе описаны абстрактный класс A и производный от этого класса класс A1. Какая из записей заведомо неверная?

- A. A * a = new A; +
- B. A1 a1;
- C. A1 a1; A & a = a1;
- D. A1 a1; A1 a2;

9. Укажите правильный вариант доступа к членам объектов (язык C ++), описанных следующим образом: `class my {char s; public: double Z; int f (int c, int d) {return c + d;}; } T1, T2;`

- A. T1.Z = 23.1; +
- B. T2-> f (2,1);
- C. T1.s = '#';
- D. my.T2-> s = 'L';

10. Какой из вариантов записи абстрактного класса в C ++ является правильным?

- A. `abstract class A {virtual int f () = 0;};`
- B. `class A {virtual int f () = 0;}; +`
- C. `class A {virtual int f () = 0;} abstract;`
- D. `class A {virtual int f ();};`

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

7.2.5 Примерный перечень вопросов для подготовки к экзамену

1. Понятие класса в C++. Применение конструкторов и деструкторов класса.
2. Перегруженные и встраиваемые (inline) функции в C++.
3. Области видимости переменных и методов в классе, использование статических (static) объектов.
4. Наследование в C++. Примеры простого и множественного наследования.
5. Виртуальные функции. Примеры.
6. Работа с шаблонными методами и классами в C++.
7. Понятие пользовательского интерфейса, критерии и свойства при разработке приложений.
8. Типы пользовательских интерфейсов (CLI и GUI), основные преимущества и области применения.

9. Использование специализированных технологий при создании приложений.
Пример программы с описанием всех файлов проекта.
10. Среда разработки (IDE) (работа с листингом, отладчиком, настройка и сборка проекта).
11. Использование визуальных средств разработки, в которых реализован GUI интерфейс.
12. Механизм обработки сообщений в разрабатываемом проекте. Примеры.
13. Описание базовых классов для создания окон и диалогов в приложении.
14. Разработка SDI приложения.
15. Работа с файлами.
16. Работа с датой.
17. Работа с временем.
18. Механизм наследования классов. Показать на примере с произвольным классом.
19. Классы для вывода информационных сообщений. Примеры использования.
20. Классы по работе со строками. Примеры использования.
21. Классы для ввода текста. Примеры использования.
22. Классы для отображения графической и текстовой информации. Примеры использования.
23. Классы для информирования пользователя о процессе работы приложения.
Примеры использования.
24. Классы по работе с палитрой цветов. Примеры использования.
25. Разработка собственных диалоговых окон. Модальный и немодальный режим.
26. Классы для создания кнопок. Примеры использования.
27. Контейнерные классы.
28. Структура проектного файла простейшего графического приложения.
29. Использование классов, предоставляющих подсказки пользователям в приложениях.
30. Отладочные средства IDE среды при разработке приложений.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Оценка при проведении промежуточной аттестации учитывает результаты тестирования. Экзамен проводится по экзаменационным билетам, каждый из которых содержит два теоретических вопроса и одно практическое задание, тестирование предполагает получение ответов на 10 вопросов.

1. Оценка «Неудовлетворительно» ставится в случае, если студент дал неправильные ответы на два экзаменационных вопроса, не решил практическое задание и ответил менее чем на 60% тестовых вопросов.

2. Оценка «Удовлетворительно» ставится в случае, если студент неправильно ответил на один экзаменационный вопрос или на все вопросы дал неполные ответы, не решил полностью практическое задание и ответил на 60-80 % тестовых вопросов.

3. Оценка «Хорошо» ставится в случае, если студент правильно ответил на один экзаменационный вопрос или на все вопросы дал неполные ответы, решил полностью практическое задание, продемонстрировал понимание материала, но допустил незначительные ошибки, а также выполнил тест на 80-90%.

4. Оценка «Отлично» ставится, если студент дал обоснованные, глубокие и теоретически правильные ответы на экзаменационные вопросы, решил полностью практическое задание, продемонстрировал понимание материала, а также выполнил тест на 90-100%.

Компетенции считаются сформированными, если в ходе изучения дисциплины выполнены и защищены курсовой проект и лабораторные работы. Выполненные лабораторные работы являются допуском к сдаче экзамена.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Интегрированные среды и средства разработки на языке программирования высокого уровня	ОПК-8, ОПК-9	Тест, экзамен, защита лабораторных работ, защита курсового проекта
2	Базовые конструкции и ключевые слова языка программирования высокого уровня	ОПК-8, ОПК-9	Тест, экзамен, защита лабораторных работ, защита курсового проекта
3	Объектно-ориентированный подход к разработке программных средств	ОПК-8, ОПК-9	Тест, экзамен, защита лабораторных работ, защита курсового проекта
4	Разработка приложений с графическими интерфейсными возможностями	ОПК-8, ОПК-9	Тест, экзамен, защита лабораторных работ, защита курсового проекта
5	Создание подсистем на основе подключаемых библиотек	ОПК-8, ОПК-9	Тест, экзамен, защита лабораторных работ, защита курсового проекта

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 45 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется с использованием выданных задач на бумажном носителе. Время решения задач 45 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется с использованием выданных задач на бумажном носителе. Время решения задач 45 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсового проекта осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8. УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Перечень учебной литературы, необходимой для освоения дисциплины

Холопкина Л.В. Программирование на C++. Учебное пособие / Л.В. Холопкина. – Воронеж, ВГТУ. – 2011. – 184 с.

Ганцева Е.А. Разработка приложений в Visual Studio: лабораторный практикум: учеб. пособие / Е.А. Ганцева. – Воронеж: ВГТУ. – 130 с.

Холопкина Л.В. Сборник задач по дисциплине “Объектно-ориентированное программирование” / Л.В. Холопкина, Н.И. Гребенникова, М.П. Носачева. – Воронеж: ВГТУ, – 2012.

Лисицин Д.В. Объектно-ориентированное программирование [Электронный ресурс]: конспект лекций/ Д.В. Лисицин— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2010.— 88 с.— Режим доступа: <http://www.iprbookshop.ru/44970.html>.— ЭБС «IPRbooks».

Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ П.В. Новиков— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 с.— Режим доступа: <http://www.iprbookshop.ru/64650.html>.— ЭБС «IPRbooks».

8.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Microsoft Windows 7, Microsoft Visual Studio 2010, Microsoft Office 2013 (Word, Access, Excel, PowerPoint, Visio)

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения лекционных занятий необходима аудитория, оснащенная проектором

Проведение лабораторных работ проводятся в специализированной лаборатории

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Объектно-ориентированное программирование» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все воз-

	возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начинаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.