

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 43

УТВЕРЖДАЮ
Руководитель образовательной программы

доц., к.т.н.
(должность, уч. степень, звание)

А.А. Фоменкова
(инициалы, фамилия)

(подпись)
«06» февраля 2025 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Объектно-ориентированное программирование»
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.04
Наименование направления подготовки/ специальности	Программная инженерия
Наименование направленности	Проектирование программных систем
Форма обучения	очная
Год приема	2025

Лист согласования рабочей программы дисциплины

Программу составил (а)

ст. преподаватель
(должность, уч. степень, звание)

(подпись, дата)
05.02.2025

Е.О. Шумова
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«06» февраля 2025 г, протокол № 01/2025

Заведующий кафедрой № 43

д.т.н., проф.
(уч. степень, звание)

(подпись, дата)
06.02.2025

М.Ю. Охтилев
(инициалы, фамилия)

Заместитель директора института №4 по методической работе

доц., к.т.н.
(должность, уч. степень, звание)

(подпись, дата)
06.02.2025

А.А. Фоменкова
(инициалы, фамилия)

Аннотация

Дисциплина «Объектно-ориентированное программирование» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.04 «Программная инженерия» направленности «Проектирование программных систем». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-2 «Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности»

ОПК-4 «Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью»

ОПК-6 «Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов»

Содержание дисциплины охватывает круг вопросов, связанных с изучением, разработкой, отладкой, тестированием объектно-ориентированного кода.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа обучающегося, курсовое проектирование.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме дифференцированного зачета.

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц, 216 часов.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

- получение студентами необходимых базовых навыков в области программирования на языках высокого уровня;
- формирование профессиональной подготовки в области разработки объектно-ориентированных программ.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Общепрофессиональные компетенции	ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.3.1 знает и понимает принципы работы современных информационных технологий, включая системы искусственного интеллекта, и программных средств, в том числе отечественного производства
Общепрофессиональные компетенции	ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.3.1 знает основные стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы
Общепрофессиональные компетенции	ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и	ОПК-6.3.1 знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий ОПК-6.У.1 умеет применять языки программирования и работы с базами данных, современные программные

	программирования к проектированию, конструированию и тестированию программных продуктов	среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ ОПК-6.В.1 имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач
--	---	--

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Информатика»,
- «Основы программирования»,
- «Алгоритмы и структуры данных».

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- «Проектирование баз данных»,
- «Проектирование программных систем»,
- «Защита информации».

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам	
		№4	№5
1	2	3	4
Общая трудоемкость дисциплины, ЗЕ/ (час)	6/ 216	4/ 144	2/ 72
Из них часов практической подготовки			
Аудиторные занятия, всего час.	85	68	17
в том числе:			
лекции (Л), (час)	34	34	
практические/семинарские занятия (ПЗ), (час)			
лабораторные работы (ЛР), (час)	34	34	
курсовой проект (работа) (КП, КР), (час)	17		17
экзамен, (час)			
Самостоятельная работа, всего (час)	131	76	55
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Дифф. Зач.,	Дифф. Зач.	

Примечание: ** кандидатский экзамен

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.
Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 4					
Раздел 1. Основные концепции объектно-ориентированного программирования	8		4		20
Раздел 2. Основы QT	14		16		30
Раздел 3. Паттерны проектирования	12		14		26
Итого в семестре:	34		34		76
Семестр 5					
Выполнение курсового проекта				17	
Итого в семестре:				17	55
Итого	34	0	34	17	131

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	<p>Основные концепции объектно-ориентированного программирования.</p> <p>1.1. Определение ООП и его основные концепции: абстракция, инкапсуляция, наследование, полиморфизм.</p> <p>1.2. Классы. Синтаксис объявления. Модификаторы доступа. Сетторы и гетторы. Создание экземпляров классов, ссылки и указатели на объекты. Статические и нестатические члены класса.</p> <p>1.3. Конструкторы. Назначение конструкторов. Конструктор по умолчанию, копирования, с параметрами. Деструкторы. Назначение деструкторов.</p>
2	<p>Основы QT</p> <p>2.1. Обзор иерархии классов Qt</p> <p>2.2. Механизм сигналов и слотов.</p> <p>2.3. Интегрированная среда разработки</p> <p>2.4. Библиотека контейнеров. Последовательные и ассоциативные контейнеры.</p> <p>2.5. Библиотека контейнеров. Алгоритмы, применяемые к контейнерам.</p> <p>2.6. Библиотека контейнеров. Класс QString. Использование</p>

	регулярных выражений для обработки строк. 2.7. Интервью или модель управления. Создание собственных представлений Qt
3	Паттерны проектирования 3.1. Диаграмма классов UML. Взаимосвязи между классами. Реализация взаимосвязей. 3.2. Введение в паттерны проектирования. 3.3. Порождающие шаблоны проектирования. Паттерн «фабричный метод» 3.4. Структурные шаблоны проектирования. 3.5. Поведенческие шаблоны проектирования.

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Учебным планом не предусмотрено					
Всего					

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 4				
1.	Классы. Определение методов класса. Права доступа.	4		1
2.	Знакомство со средой Qt Creator. Создание приложения в QtDesigner с использованием виджетов QLabel, QTextEdit, QPushButton	2		2
3.	Создание приложения с использованием виджетов QComboBox	2		2
4.	Создание приложения с использованием QCheckBox	2		2
5.	Разработка программы с использованием контейнеров библиотеки Qt	4		2
6.	Разработка программы с использованием порождающих шаблонов проектирования	4		2,3
7.	Разработка программы с использованием	4		2,3

	структурных шаблонов проектирования в QtDesigner			
8.	Разработка программы с использованием поведенческих шаблонов проектирования в QtDesigner	4		2,3
9.	Разработка программы с реализацией многозначной зависимости с использованием шаблонов проектирования в QtDesigner	8		3
Всего		34		

4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсового проекта: приобретение практических навыков применения моделей и методов объектно-ориентированной парадигмы, разработки и отладки программного обеспечения. Закрепление навыков работы с инструментальными программными средствами.

Примерные темы заданий на курсовой проект приведены в разделе 10 РПД.

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 4, час	Семестр 5, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)	50	40	10
Курсовое проектирование (КП, КР)	40		40
Подготовка к текущему контролю успеваемости (ТКУ)	20	20	
Подготовка к промежуточной аттестации (ПА)	21	16	5
Всего:	131	76	55

5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
http://e.lanbook.com/book/90158	Конова, Е.А. Алгоритмы и программы. Язык C++. [Электронный ресурс] / Е.А. Конова, Г.А. Поллак. — Электрон. дан. —	

	СПб. : Лань, 2017. — 384 с.	
004.432-B19	Васильев, А. Н. Самоучитель С++ с примерами и задачами : [учебное пособие] / А. Н. Васильев. - СПб. : Наука и техника, 2010. - 480 с.	Отдел фундаментальной литературы – 30
004.43(075)	Объектно-ориентированное программирование: учеб.пособие/ Е.О..Шумова.-ГУАП,2021.-115 с.	50
http://e.lanbook.com/book/1220	Приемы объектно-ориентированного проектирования. Паттерны проектирования = Design Patterns. Elements of reusable object-oriented software/ Э. Гамма [идр.] ;пер. сангл. А. Слинкин. - СПб.: ПИТЕР, 2008. - 366 с.Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно ориентированного проектирования. Паттерны проектирования [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2007. — 367с.	
http://e.lanbook.com/book/5115	Ашарина И.В. Объектно-ориентированное программирование в С++: лекции и упражнения. [Электронный ресурс] — Электрон. дан. — М. : "Горячая линия-Телеком", 2012. — 320 с.	

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
https://e.lanbook.com/	Электронная библиотечная система Лань (доступ для обучающихся и работников ГУАП)
https://znanium.com/	Электронная библиотечная система Знаниум (доступ для обучающихся и работников ГУАП)
https://urait.ru/	Образовательная платформа Юрайт (доступ для обучающихся и работников ГУАП)
https://ravesli.com/uroki-cpp/	Уроки программирования на языке С++
http://cppstudio.com/cat/274/	Язык программирования С++

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10 – Перечень программного обеспечения

№ п/п	Наименование
1	Операционная система Microsoft Windows 10
2	Microsoft Visual Studio
3	Qt-Creator (распространяется бесплатно)
4	Средства демонстрации PDF файлов
5	Единая электронная образовательная среда ГУАП
6	Microsoft Office

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	
2	Учебная аудитория для проведения лабораторных занятий оснащенная специализированной мебелью; техническими средствами обучения, служащими для представления учебной информации большой аудитории; лабораторным оборудованием (ПЭВМ, объединенных в локальную вычислительную сеть с выходом в вычислительную сеть ГУАП и Интернет)	ул. Гастелло, д. 15, лит. А, ауд. 24-03, 24-05; ул. Б.Морская, д. 67, лит. А, ауд. 23-08, 23-09, 23-10

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Дифференцированный зачёт	Список вопросов; Тесты; Задачи.
Выполнение курсового проекта	Экспертная оценка на основе требований к содержанию курсового проекта.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться

100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 – Критерии оценки уровня сформированности компетенций

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
«отлично» «зачтено»	<ul style="list-style-type: none"> – обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий.
«хорошо» «зачтено»	<ul style="list-style-type: none"> – обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий.
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> – обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий.
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> – обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
	Учебным планом не предусмотрено	

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
1.	Реализация классов в C++. Класс, структура, объединение (class, struct, union) основные отличия.	ОПК-2.3.1 ОПК-6.У.1

2.	Исключения. Реализация обработки исключительных ситуаций. Вложенные контролируемые блоки операций.	ОПК-6.У.1
3.	Исключения. Классы стандартной библиотеки для обработки исключений.	ОПК-6.В.1
4.	Стандартная библиотека C++. Контейнеры (понятие), принципы использования.	ОПК-6.В.1
5.	Стандартная библиотека C++. Итераторы (понятие), виды.	ОПК-4.3.1 ОПК-6.В.1
6.	Стандартная библиотека C++. Поток (thread).	ОПК-6.В.1
7.	Шаблоны проектирования (Design patterns). Виды.	ОПК-2.3.1 ОПК-4.3.1
8.	Порождающие шаблоны. Синглтон (Singleton).	ОПК-6.В.1
9.	Порождающие шаблоны. Фабричный метод (Factory method).	ОПК-4.3.1 ОПК-6.В.1
10.	Порождающие шаблоны. Абстрактная фабрика (Abstract factory).	ОПК-6.В.1
11.	Порождающие шаблоны. Строитель (Builder).	ОПК-4.3.1 ОПК-6.В.1
12.	Порождающие шаблоны. Прототип (Prototype).	ОПК-4.3.1 ОПК-6.В.1
13.	Структурные шаблоны. Адаптер (Adapter).	ОПК-4.3.1 ОПК-6.В.1
14.	Структурные шаблоны. Мост (Bridge).	ОПК-4.3.1 ОПК-6.У.1 ОПК-6.В.1
15.	Структурные шаблоны. Компонент (Composite).	ОПК-4.3.1 ОПК-6.У.1 ОПК-6.В.1
16.	Структурные шаблоны. Декоратор (Decorator).	ОПК-6.В.1
17.	Структурные шаблоны. Фасад (Facade).	ОПК-4.3.1 ОПК-6.У.1 ОПК-6.В.1
18.	Структурные шаблоны. Прокси (Proxy).	ОПК-4.3.1 ОПК-6.В.1
19.	Структурные шаблоны. Приспособленец (Flyweight).	ОПК-4.3.1 ОПК-6.В.1
20.	Поведенческие шаблоны. Команда (Command).	ОПК-4.3.1 ОПК-6.В.1
21.	Поведенческие шаблоны. Итератор (Iterator).	ОПК-4.3.1 ОПК-6.В.1
22.	Поведенческие шаблоны. Посредник (Mediator).	ОПК-4.3.1 ОПК-6.У.1 ОПК-6.В.1
23.	Поведенческие шаблоны. Хранитель (Memento).	ОПК-4.3.1 ОПК-6.В.1
24.	Поведенческие шаблоны. Наблюдатель (Observer).	ОПК-4.3.1 ОПК-6.В.1
25.	Поведенческие шаблоны. Состояние (State).	ОПК-4.3.1 ОПК-6.В.1
26.	Поведенческие шаблоны. Стратегия (Strategy).	ОПК-6.В.1
27.	Поведенческие шаблоны. Шаблонный метод (Template)	ОПК-6.В.1

	method).	
28.	Поведенческие шаблоны. Посетитель (Visitor).	ОПК-6.B.1
29.	Поведенческие шаблоны. Цепочка обязанностей (Chain of Responsibility).	ОПК-6.B.1

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
1	Разработка иерархии классов для распределения учебной нагрузки преподавателям кафедры (В программе распределение осуществляется на основе данных по количеству групп, учебного плана для конкретной группы, закрепления дисциплин за преподавателями и количества часов, которые должен отработать преподаватель)
2	Разработка иерархии классов для составления расписания учебных занятий в ВУЗе. (Расписание составляется на основе распределения нагрузки из темы 1)
3	Разработка системы классов, описывающих различные транспортные средства, реализуемые дилерским центром. (В программе учитываются закупки того или иного бренда от производителя, далее продажа покупателям с фиксацией всех данных о покупках-продажах и данных покупателя)
4	Разработка иерархии классов для обеспечения работы библиотеки (В программе должны поддерживаться режимы поиска книги по заданному критерию (автор, название), заказа книги, учета клиентов и книг в книгохранилище, выдачи отчетов по запросам (местонахождение книги в архиве или ее отсутствие)).
5	Разработка иерархии классов для обеспечения работы магазина (В программе должны поддерживаться режимы заказа товара (продовольственных товаров), покупки и учета товаров, анализа востребованности товара в зависимости от времени дня и дня недели).
6	Разработка иерархии классов для обеспечения работы гостиницы (В программе должны поддерживаться режимы учета и распределения по номерам приезжих гостей в зависимости от требований и пожеланий проживающих).
7	Разработка иерархии классов для обеспечения работы театра (В программе должны поддерживаться режимы учета спектаклей, актеров, играющих в спектаклях, концертов, распределения мест и стоимости билетов, анализа популярности спектаклей по различным критериям).
8	Разработка иерархии классов «Успеваемость студентов на факультете» (В программе должны поддерживаться режимы учета учащихся и результатов сдачи экзаменов, анализа сессии по семестрам, по факультетам, специальностям, генерации отчетов отличников и двоечников.)
9	Разработка иерархии классов для обеспечения работы центра занятости (В программе должны поддерживаться режимы учета безработных,

	<i>их стажа, квалификации, желания работать по определенной специальности, места расположения и заработной платы, учета уже стоящих на учете в центре занятости, анализа занятости от времени, специальности и т.д.)</i>
10	<i>Разработка иерархии классов для обеспечения работы больницы (В программе должны поддерживаться режимы учета больных по отделениям, заболеваниям, сложности заболевания и количеству заболеваний у одного человека, продолжительности болезни, количеству койко-мест и анализа заболеваемости по районам и категориям)</i>

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора								
1	Инструкция: выберите один правильный ответ Что такое конструктор класса в ООП? а) Метод, который вызывается при создании объекта класса и инициализирует его свойства б) Метод, который вызывается при уничтожении объекта класса и освобождает занятые им ресурсы с) Метод, который определяет новое поведение для стандартных операторов д) Метод, который скрывает реализацию объекта от пользователя	ОПК-2.3.1								
2	Инструкция: Прочитайте текст, выберите правильные варианты ответа При использовании копирующего конструктора: 1) Вновь созданный объект зависит от копируемого 2) Вновь созданный объект не зависит от копируемого 3) Вновь созданный объект имеет те же атрибуты что и исходный	ОПК-2.3.1								
3.	Инструкция: Прочитайте текст и установите соответствие. К каждой позиции, данной в левом столбце, подберите соответствующую позицию в правом столбце <table><tr><td>A</td><td>Модификатор доступа, обозначающий защищенные члены класса</td><td>1</td><td>static</td></tr><tr><td>B</td><td>Модификатор доступа, обозначающий статические</td><td>2</td><td>public</td></tr></table>	A	Модификатор доступа, обозначающий защищенные члены класса	1	static	B	Модификатор доступа, обозначающий статические	2	public	ОПК-2.3.1
A	Модификатор доступа, обозначающий защищенные члены класса	1	static							
B	Модификатор доступа, обозначающий статические	2	public							

	<table><tr><td></td><td>класса</td><td></td><td></td></tr><tr><td>C</td><td>Модификатор доступа, обозначающий закрытые члены класса</td><td></td><td>protected</td></tr><tr><td>D</td><td>Модификатор доступа, обозначающий открытые члены класса</td><td></td><td>private</td></tr></table>		класса			C	Модификатор доступа, обозначающий закрытые члены класса		protected	D	Модификатор доступа, обозначающий открытые члены класса		private	
	класса													
C	Модификатор доступа, обозначающий закрытые члены класса		protected											
D	Модификатор доступа, обозначающий открытые члены класса		private											
4	<p>Инструкция: Проанализируйте программный код и запишите в ответе что будет получено в результате работы этого кода:</p> <pre>struct Animal { string tip; string name; }; int main() { setlocale(LC_ALL, "rus"); system("color F1"); Animal z1, z2; z1.tip = "Кот"; z1.name = "Базилио"; z2.tip = "Лиса"; z2.name = "Алиса"; z1 = z2; z2.tip = "Заяц"; cout << z1.tip << " " << z1.name << endl; cout << z2.tip << " " << z2.name << endl; }</pre>	ОПК-2.3.1												
5	<p>Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ</p> <p>Что такое объект в ООП?</p>	ОПК-2.3.1												
6	<p>Инструкция: Проанализируйте программный код и запишите в ответе что будет получено в результате работы этого кода:</p> <p>Что выведет на экран следующий код?</p> <pre>#include <iostream> using namespace std; class Base { public: virtual void print() { cout << "Base class" << endl; } };</pre>	ОПК-2.3.1												

	<pre>class Derived : public Base { public: void print() { cout << "Derived class" << endl; } }; int main() { Base* base = new Derived(); Derived* derived = dynamic_cast<Derived*>(base); derived->print(); delete base; return 0; }</pre>									
7	<p>Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ</p> <p>Что такое дружественный класс в ООП на языке C++?</p>	ОПК-2.3.1								
8	<p>Инструкция: выберите один правильный ответ</p> <p>Что такое объект в ООП?</p> <p>а) Шаблон, описывающий свойства и методы класса б) Экземпляр класса, имеющий конкретные значения свойств с) Метод класса, реализующий определенную функцию д) Переменная, хранящая значение определенного типа данных</p>	ОПК-4.3.1								
9	<p>Инструкция: Прочитайте текст, выберите правильные варианты ответа</p> <p>Имеется абстрактный класс А и класс А1, являющийся производным от класса А. Какие из следующих записей заведомо неверны?</p> <p>1) А1 a1; А &a=a1; 2) А *a=new А; 3) А a;</p>	ОПК-4.3.1								
10	<p>Инструкция: Для каждой элементарной структуры данных, указанной в левом столбце, подберите соответствующую область допустимых значений, указанную в правом столбце</p> <table><tr><td>А</td><td>Механизм программирования, объединяющий вместе код и данные, которыми он манипулирует</td><td>1</td><td>Метод</td></tr><tr><td>В</td><td>Процесс, в ходе которого один объект приобретает свойства другого</td><td>2</td><td>Инкапсуляция</td></tr></table>	А	Механизм программирования, объединяющий вместе код и данные, которыми он манипулирует	1	Метод	В	Процесс, в ходе которого один объект приобретает свойства другого	2	Инкапсуляция	ОПК-4.3.1
А	Механизм программирования, объединяющий вместе код и данные, которыми он манипулирует	1	Метод							
В	Процесс, в ходе которого один объект приобретает свойства другого	2	Инкапсуляция							

	<table><tr><td>C</td><td>Позволяет одному интерфейсу получить доступ к общему классу действий</td><td>3</td><td>Наследование</td></tr><tr><td>D</td><td>Подпрограмма, которая манипулирует данными, определенными в классе</td><td>4</td><td>Полиморфизм</td></tr></table>	C	Позволяет одному интерфейсу получить доступ к общему классу действий	3	Наследование	D	Подпрограмма, которая манипулирует данными, определенными в классе	4	Полиморфизм	
C	Позволяет одному интерфейсу получить доступ к общему классу действий	3	Наследование							
D	Подпрограмма, которая манипулирует данными, определенными в классе	4	Полиморфизм							
11	Инструкция: Проанализируйте фрагмент программного кода. В ответе запишите, что выведет следующий код при создании экземпляра класса D? <pre>struct A { A() { cout << "A"; } }; struct B : virtual A { B() { cout << "B"; } }; struct C : virtual A { C() { cout << "C"; } }; struct D : B, C { D() { cout << "D"; } };</pre>	ОПК-4.3.1								
12	Инструкция: Проанализируйте программный код и запишите развернутый обоснованный ответ: Вызовет ли данный код ошибку компиляции? <pre>class Rectangle { public int a,b; int sum(); int square(); ~rect(); };</pre>	ОПК-4.3.1								
13	Инструкция: Прочитайте вопрос и запишите развернутый обоснованный ответ Что такое статический метод в ООП на языке C++?	ОПК-4.3.1								
14	Инструкция: Проанализируйте программный код и запишите в ответе что будет получено в результате работы этого кода: <pre>class A { public: A() { f(); } virtual void f() { std::cout << "A::f"; } }; class B : public A {</pre>	ОПК-4.3.1								

	<pre>public: void f() { std::cout << "B::f"; } }; int main(int argc, char * argv[]) { A * a = new B(); delete a; return 0; }</pre>									
15	<p>Инструкция: выберите один правильный ответ Что такое инкапсуляция в ООП?</p> <p>а) Возможность создания нескольких методов с одним и тем же именем, но различными параметрами б) Механизм, позволяющий скрыть реализацию объекта от пользователя в) Возможность объектов с одинаковым интерфейсом иметь различную реализацию г) Механизм, позволяющий создавать новый класс на основе уже существующего</p>	ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1								
16	<p>Инструкция: Прочитайте текст, выберите правильные варианты ответа и запишите аргументы, обосновывающие выбор ответов.</p> <p>Какие ключевые слова используются для создания и обработки исключительных ситуаций:</p> <p>1) catch 2) delete 3) try 4) throw</p>	ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1								
17	<p>Инструкция: Для каждой элементарной структуры данных, указанной в левом столбце, подберите соответствующую область допустимых значений, указанную в правом столбце</p> <table><tr><td>А</td><td>Конструктор</td><td>1</td><td>Вызывается автоматически, как только объект класса уничтожается</td></tr><tr><td>В</td><td>Деструктор</td><td>2</td><td>Имеет доступ к защищенным и собственным членам класса, не</td></tr></table>	А	Конструктор	1	Вызывается автоматически, как только объект класса уничтожается	В	Деструктор	2	Имеет доступ к защищенным и собственным членам класса, не	ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
А	Конструктор	1	Вызывается автоматически, как только объект класса уничтожается							
В	Деструктор	2	Имеет доступ к защищенным и собственным членам класса, не							

				являясь его компонентом		
	C	Дружественная функция	3	Возможность распространения стандартных операций на операнды, определенные пользователем		
	D	Перегруженный оператор	4	Используется для инициализации объектов класса		
18	Инструкция: Проанализируйте программный код и запишите в ответе что будет получено в результате работы этого кода: Что напечатает следующий код при создании экземпляра класса X: <pre>class Y { public: Y() { cout << "Y"; } }; class Z { public: Z() { cout << "Z"; } }; class X : public Z { private: Y m_objY; public: X() { cout << "X"; } }; .</pre>					ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
19	Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ Что такое композиция в ООП на языке C++?					ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
20	Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ Что такое константный метод в ООП на языке C++?					ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в

локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины (Ниже приводятся рекомендации по составлению данного раздела)

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

11.2. Методические указания для обучающихся по участию в семинарах *(не предусмотрено учебным планом по данной дисциплине)*

11.3. Методические указания для обучающихся по прохождению практических занятий *(не предусмотрено учебным планом по данной дисциплине)*

11.4. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

задание на лабораторную работу загружается в личный кабинет. Для проведения лабораторных работ необходим компьютерный класс с установленным программным обеспечением, перечисленным в таблице 10.

Структура и форма отчета о лабораторной работе

- студент выполняет лабораторную работу (согласно варианта задания);
- демонстрирует работу программы преподавателю;
- загружает отчет в личный кабинет;
- защищает работу.

Требования к оформлению отчета о лабораторной работе

отчет о работе оформляется в виде *.doc или *.pdf-файла. Имя файла с отчетом должно содержать № группы, фамилию студента, № работы (например, 4531_Иванов_ЛР1).

Отчет включает в себя:

- титульный лист;
- условие (с указанием номера варианта);
- полный текст (листинг) программы;
- на скриншотах должны быть отображены исходные данные и результаты расчетов (скриншоты должны демонстрировать все возможные ветви алгоритма решения);
- вывод.

11.5. Методические указания для обучающихся по прохождению курсового проектирования/выполнения курсовой работы

Курсовой проект/ работа проводится с целью формирования у обучающихся опыта комплексного решения конкретных задач профессиональной деятельности.

Курсовой проект/ работа позволяет обучающемуся:

- закрепить знания языка программирования и современной программной среды разработки информационных систем и технологий;
- закрепить навыки программирования, отладки и тестирования программного обеспечения;
- научиться оформлять пояснительную записку к курсовому проекту.

Структура пояснительной записки курсового проекта/ работы

- Титульный лист
- Задание на курсовое проектирование
- Содержание
- Постановка задачи
- Проектирование классов
- Разработка приложения
- Тестирование
- Заключение

- Список использованных источников
- Приложения.

Требования к оформлению пояснительной записки курсового проекта/ работы

Текст пояснительной записки разделяют на разделы, подразделы и пункты.

Разделы должны иметь порядковые номера в пределах всей пояснительной записки, обозначенные арабскими цифрами без точки, и начинаться с абзацного отступа.

Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номеров раздела и подраздела, разделенных точкой. В конце номера подраздела точка не ставится. Например: 2.1 – первый подраздел второго раздела.

Разделы, как и подразделы, могут состоять из одного или нескольких пунктов.

Подробные методические указания по оформлению курсового проекта приведены в методических указаниях к выполнению курсового проекта.

11.6. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся является учебно-методический материал по дисциплине.

11.7. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

В течение семестра №4 текущий контроль успеваемости выполняется по результатам защиты лабораторных работ. Требования к положительному оцениванию текущей успеваемости предусматривают обязательное выполнение всех лабораторных работ в указанные календарные сроки. Результаты текущего контроля успеваемости учитываются при проведении промежуточной аттестации.

11.8. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– дифференцированный зачет – это форма оценки знаний, полученных обучающимся при изучении дисциплины.

Студент допускается к сдаче дифференцированного зачета только при условии выполнения всех обязательных заданий, предусмотренных текущим контролем.

Дифференцированный зачет может проводиться путем тестирования или устного опроса. Вопросы для устного опроса приведены в таблице 16, а для тестирования – в таблице 18.

За выполненные в течение семестра работы студент накапливает баллы, которые также будут учтены при проведении промежуточной аттестации.

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой