

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

Кафедра № 43

УТВЕРЖДАЮ

Руководитель направления

доц., к.т.н., доц.

(должность, уч. степень, звание)

В.А. Матьяш

(инициалы, фамилия)

(подпись)

«15» июня 2022 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Объектно-ориентированное программирование»

(Наименование дисциплины)

Код направления подготовки/ специальности	02.03.03
Наименование направления подготовки/ специальности	Математическое обеспечение и администрирование информационных систем
Наименование направленности	Системный анализ в информационных технологиях
Форма обучения	очная

Лист согласования рабочей программы дисциплины

Программу составил (а)

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

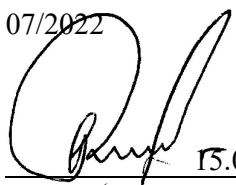
 15.06.2022  
(подпись, дата)

Е.О.Шумова  
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43  
«15» июня 2022 г., протокол № 07/2022

Заведующий кафедрой № 43


д.т.н.,проф.  
(уч. степень, звание)

 15.06.2022  
(подпись, дата)

М.Ю. Охтилев  
(инициалы, фамилия)

Ответственный за ОП ВО 02.03.03(02)

старший преподаватель  
(должность, уч. степень, звание)

 15.06.2022  
(подпись, дата)

А.А. Фоменкова  
(инициалы, фамилия)

Заместитель директора института №4 по методической работе

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

 15.06.2022  
(подпись, дата)

А.А. Ключарев  
(инициалы, фамилия)

## Аннотация

Дисциплина «Объектно-ориентированное программирование» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 02.03.03 «Математическое обеспечение и администрирование информационных систем» направленности «Системный анализ в информационных технологиях». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-2 «Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений»

ОПК-3 «Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения»

ОПК-4 «Способен участвовать в разработке технической документации программных продуктов и программных комплексов»

ОПК-6 «Способен использовать в педагогической деятельности научные основы знаний в сфере информационно-коммуникационных технологий»

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 экзамена.

Общая трудоемкость освоения дисциплины составляет 7 зачетных единиц, 252 часа.

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
Универсальные компетенции	УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.В.3 владеть навыками использования цифровых средств для решения поставленной задачи
Общепрофессиональные компетенции	ОПК-3 Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	ОПК-3.3.1 знает основные положения и концепции в прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов. ОПК-3.У.1 умеет использовать современные информационные технологии в профессиональной деятельности при создании программных продуктов и программных комплексов различного назначения. ОПК-3.В.1 имеет практические навыки разработки программного обеспечения.
Общепрофессиональные компетенции	ОПК-4 Способен участвовать в разработке технической документации программных	ОПК-4.В.1 имеет практические навыки подготовки технической документации

	продуктов и программных комплексов	
Общепрофессиональные компетенции	ОПК-6 Способен использовать в педагогической деятельности научные основы знаний в сфере информационно-коммуникационных технологий	ОПК-6.3.1 знает изучаемый язык программирования, сетевые технологии, применение веб-технологий

## 2. Место дисциплины в структуре ОП

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

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

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

- Web-технологии»,
- «Проектирование программных систем»,
- «Защита информации»,
- «Компьютерная графика».

## 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
<b>Общая трудоемкость дисциплины, ЗЕ/ (час)</b>	7/ 252	5/ 180	2/ 72
<b>Из них часов практической подготовки</b>			
<b>Аудиторные занятия, всего час.</b>	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)	36	36	
<b>Самостоятельная работа, всего (час)</b>	131	76	55
<b>Вид промежуточной аттестации:</b> зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз.,	Экз.	

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	12		14		30
Раздел 3. Паттерны проектирования	14		16		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 style="text-align: center;"><b>Основные концепции объектно-ориентированного программирования.</b></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><b>Основы QT</b></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</p>

	ассоциативные контейнеры. 2.5. Библиотека контейнеров. Алгоритмы, применяемые к контейнерам. 2.6. Библиотека контейнеров. Класс QString. Использование регулярных выражений для обработки строк. 2.7. Классы-виджеты Qt.
<b>3</b>	<b>Паттерны проектирования</b> 2.1. Введение в паттерны проектирования. 2.2. Порождающие шаблоны проектирования. Паттерн «фабричный метод» 2.3. Паттерны «прототип» и «синглтон». 2.4. Паттерны «абстрактная фабрика» и «строитель». 2.5. Структурные шаблоны проектирования. Паттерны «адаптер», «мост» и «компоновщик». 2.6. Паттерны «декоратор», «фасад», «приспособленец» и «прокси». 2.7. Поведенческие шаблоны проектирования. Паттерны команда», «итератор». «посредник», и «хранитель». 2.8. Паттерны «наблюдатель», «состояние» 2.9. Паттерны «стратегия», «шаблонный метод». 2.10. Паттерны «посетитель» и «цепочка обязанностей».

#### 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	1
2.	Знакомство со средой Qt Creator. Создание консольного приложения.	2	1	2
3.	Создание приложения в QtDesigner c	2	1	2

	использованием виджетов QLabel, QTextEdit, QPushButton			
4.	Создание приложения с использованием виджетов QComboBox	2	1	2
5.	Создание приложения с использованием QCheckBox	2	1	2
6.	Создание приложения с использованием контейнеров библиотеки Qt	4	2	2
7.	Создание приложения с использованием меню и панели элементов	2	2	2
8.	Описание классов и порождение объектов с использованием шаблонов singleton, prototype.	4	2	3
9.	Разработка программы с использованием структурных шаблонов проектирования в QtDesigner	4	2	3
10.	Разработка приложения с использованием поведенческих шаблонов проектирования в QtDesigner	4	2	3
11.	Разработка приложения с использованием порождающих паттернов проектирования в QtDesigner	4	2	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)	46	36	10
Курсовое проектирование (КП, КР)	40		40
Подготовка к текущему контролю успеваемости (ТКУ)	20	20	
Подготовка к промежуточной аттестации (ПА)	25	20	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)
<a href="http://e.lanbook.com/book/90158">http://e.lanbook.com/book/90158</a>	Конова, Е.А. Алгоритмы и программы. Язык С++. [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
<a href="http://e.lanbook.com/book/1220">http://e.lanbook.com/book/1220</a>	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.	
<a href="http://e.lanbook.com/book/5115">http://e.lanbook.com/book/5115</a>	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
<a href="https://ravesli.com/uroki-cpp/">https://ravesli.com/uroki-cpp/</a>	Уроки программирования на языке C++
<a href="http://cppstudio.com/cat/274/">http://cppstudio.com/cat/274/</a>	Язык программирования C++

## 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	Microsoft Visual Studio
2	Qt-Creator (распространяется бесплатно)
3	ОС Windows

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 таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	
2	Компьютерный класс	

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

### 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
	ООП – определение. Основные понятия ООП: Абстракция, наследование.	УК-2.В.3
1.	ООП – определение. Основные понятия ООП: Инкапсуляция, полиморфизм.	УК-2.В.3 ОПК-2.3.1
2.	ООП – основные принципы. Определение класса, объекта. Их взаимосвязь между собой.	УК-2.В.3 ОПК-6.У.1
3.	С++ - Пространства имен. Объявление и способы использования.	ОПК-2.3.1 ОПК-4.У.1 ОПК-6.У.1
4.	С++ - классы памяти.	УК-2.В.3

5.	C++ - Универсальная инициализация. Инициализация по умолчанию базовых типов.	УК-2.В.3
6.	Реализация классов в C++. Объявление и определение класса. Данные-члены и функции-члены класса.	ОПК-2.3.1 ОПК-6.У.1
7.	Реализация классов в C++. Класс, структура, объединение (class, struct, union) основные отличия.	ОПК-2.3.1 ОПК-6.У.1
8.	Реализация классов в C++. Права доступа к членам класса.	ОПК-6.У.1
9.	Реализация классов в C++. Друзья класса. (Дружественные классы и функции).	ОПК-6.У.1
10.	Реализация классов в C++. Перегрузка функций-членов.	ОПК-6.У.1
11.	Реализация классов в C++. Статические и нестатические члены.	ОПК-6.У.1
12.	Конструкторы. Конструкторы по умолчанию.	УК-2.В.3 ОПК-6.У.1
13.	Конструкторы. Инициализирующий конструктор, список инициализации и конструктор копирования.	ОПК-6.У.1
14.	Конструкторы. Конструктор перемещения.	УК-2.В.3 ОПК-6.У.1
15.	Деструкторы. Назначение деструктора. Виртуальные деструкторы.	ОПК-6.У.1
16.	Деструкторы. Использование деструкторов. Порядок уничтожения.	ОПК-6.У.1
17.	Реализация классов в C++. Указатели на объекты. Указатель this.	ОПК-6.У.1
18.	Реализация классов в C++. Указатели на члены класса.	ОПК-6.У.1
19.	Реализация классов в C++. Шаблоны классов. Параметризация шаблонов классов.	ОПК-6.У.1
20.	Реализация классов в C++. Шаблоны классов. Специализация шаблонов классов.	ОПК-6.У.1
21.	Реализация классов в C++. Перегрузка операторов.	ОПК-6.У.1
22.	Реализация классов в C++. Операторные функции в пространстве имен и как члены класса.	ОПК-6.У.1
23.	Реализация классов в C++. Операторы ввода-вывода.	ОПК-6.У.1
24.	Реализация классов в C++. Перегрузка операторов преобразования типов.	ОПК-6.У.1
25.	Реализация классов в C++. Оператор явного (explicit) преобразования типов.	ОПК-6.У.1
26.	Реализация наследования в C++. Одиночное наследование. Область видимости членов класса.	ОПК-6.У.1
27.	Реализация наследования в C++. Множественное и виртуальное наследование.	ОПК-6.У.1
28.	Реализация полиморфизма в C++. Виртуальные функции.	ОПК-6.У.1
29.	Реализация полиморфизма в C++. Абстрактные классы. Ограничения на использование абстрактных классов.	ОПК-6.У.1
30.	Исключения. Контролируемый блок операций. Операция генерации исключения (throw).	ОПК-6.В.1
31.	Исключения. Реализация обработки исключительных ситуаций. Вложенные контролируемые блоки операций.	ОПК-6.У.1
32.	Исключения. Классы стандартной библиотеки для	ОПК-6.В.1

	обработки исключений.	
33.	C++ - тип auto.	УК-2.В.3 ОПК-6.В.1
34.	C++ - диапазонный цикл for (...). Синтаксис, применение.	ОПК-6.В.1 ОПК-2.3.1
35.	C++ - Ключевое слово decltype, назначение.	УК-2.В.3 ОПК-6.В.1
36.	C++ - Перечисления с ограниченной областью видимости.	ОПК-6.В.1
37.	Стандартная библиотека C++. Библиотека ввода-вывода. Назначение, основные классы.	УК-2.В.3 ОПК-6.В.1
38.	Стандартная библиотека C++. Контейнеры (понятие), принципы использования.	ОПК-6.В.1
39.	Стандартная библиотека C++. Итераторы (понятие), виды.	УК-2.В.3 ОПК-6.В.1
40.	Стандартная библиотека C++. Последовательные контейнеры.	ОПК-6.В.1
41.	Стандартная библиотека C++. Ассоциативные контейнеры.	ОПК-6.В.1
42.	Стандартная библиотека C++. Неупорядоченные ассоциативные контейнеры.	ОПК-6.В.1
43.	Стандартная библиотека C++. Работа со строками символов.	ОПК-6.В.1
44.	Стандартная библиотека C++. Поток (thread).	ОПК-6.В.1
45.	Шаблоны проектирования (Design patterns). Виды.	ОПК-2.3.1
46.	Порождающие шаблоны. Синглетон (Singleton).	ОПК-6.В.1
47.	Порождающие шаблоны. Фабричный метод (Factory method).	ОПК-6.В.1
48.	Порождающие шаблоны. Абстрактная фабрика (Abstract factory).	ОПК-6.В.1
49.	Порождающие шаблоны. Строитель (Builder).	ОПК-6.В.1
50.	Порождающие шаблоны. Прототип (Prototype).	ОПК-6.В.1
51.	Структурные шаблоны. Адаптер (Adapter).	ОПК-6.В.1
52.	Структурные шаблоны. Мост (Bridge).	ОПК-6.В.1
53.	Структурные шаблоны. Компоновщик (Composite).	ОПК-6.В.1
54.	Структурные шаблоны. Декоратор (Decorator).	ОПК-6.В.1
55.	Структурные шаблоны. Фасад (Facade).	ОПК-6.В.1
56.	Структурные шаблоны. Прокси (Proxy).	ОПК-6.В.1
57.	Структурные шаблоны. Приспособленец (Flyweight).	ОПК-6.В.1
58.	Поведенческие шаблоны. Команда (Command).	ОПК-6.В.1
59.	Поведенческие шаблоны. Итератор (Iterator).	ОПК-6.В.1
60.	Поведенческие шаблоны. Посредник (Mediator).	ОПК-6.В.1
61.	Поведенческие шаблоны. Хранитель (Memento).	ОПК-6.В.1
62.	Поведенческие шаблоны. Наблюдатель (Observer).	ОПК-6.В.1
63.	Поведенческие шаблоны. Состояние (State).	ОПК-6.В.1
64.	Поведенческие шаблоны. Стратегия (Strategy).	ОПК-6.В.1
65.	Поведенческие шаблоны. Шаблонный метод (Template method).	ОПК-6.В.1
66.	Поведенческие шаблоны. Посетитель (Visitor).	ОПК-6.В.1

67.	Поведенческие шаблоны. Цепочка обязанностей (Chain of Responsibility).	ОПК-6.В.1
-----	--	-----------

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

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
	Учебным планом не предусмотрено	

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, реализующих графические примитивы на плоскости с реализацией операций над множествами. ( <i>Графические примитивы – прямоугольники, окружности, эллипсы. Операции над множествами – пересечение, объединение и разность</i> )
2	Разработка иерархии классов, реализующих комплексную арифметику. ( <i>Стандартная модель комплексных чисел. Действия над комплексными числами – сложение, вычитание, умножение, деление и сравнение.</i> )
3	Разработка иерархии классов, реализующих операции над векторами. ( <i>Понятие и модель вектора. Операции над векторами – сложение, разность, произведение вектора на вещественное число.</i> )
4	Разработка системы классов для обеспечения работы деканата. Разработка системы классов для обеспечения работы с абонентами телефонной компании.
5	Разработка системы классов, описывающих различные транспортные средства, реализуемые дилерским центром.
6	Разработка системы классов для обеспечения работы библиотеки ( <i>В системе 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).</i> )
7	Разработка иерархии классов для обеспечения работы магазина ( <i>В системе должны поддерживаться режимы заказа товара (продовольственных товаров), покупки и учета товаров, анализа покупаемости товара, анализа покупаемости продуктов в зависимости от времени дня и дня недели.</i> )
8	Разработка системы классов для обеспечения работы гостиницы ( <i>в системе должны поддерживаться режимы учета и распределения по номерам приезжих гостей в зависимости от требований и пожеланий проживающих.</i> )
9	Разработка системы классов для обеспечения работы театра ( <i>В системе должны поддерживаться режимы учета спектаклей, актеров, играющих в спектаклях, концертов, распределения мест и стоимости билетов, анализа популярности спектаклей по различным критериям.</i> )
10	Разработка системы классов «Успеваемость студентов на факультете» ( <i>В системе должны поддерживаться режимы учета учащихся и результатов сдачи экзаменов, анализа сессии по семестрам, по факультетам, специальностям, генерации отчетов отличников и</i>

	<i>двоечников.)</i>
11	Разработка системы классов для обеспечения работы центра занятости <i>(В системе должны поддерживаться режимы учета безработных, 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>
12	Разработка системы классов для обеспечения работы больницы <i>(В системе 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.	<p>Что выведет следующий код при создании экземпляра класса D?</p> <pre> struct A {     A(){ cout &lt;&lt; "A "; } }; struct B : public A {     B() { cout &lt;&lt; "B "; } }; struct C : public A {     C() { cout &lt;&lt; "C "; } }; struct D : B, C {     D() { cout &lt;&lt; "D "; } };  int main(int argc, char *argv[]) {     D d;     return 0; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
2.	<pre> struct A {     ~A(){ cout &lt;&lt; "A "; } }; struct B : public A {     ~B() { cout &lt;&lt; "B "; } }; struct C : public A {     ~C() { cout &lt;&lt; "C "; } }; struct D : B, C {     ~D() { cout &lt;&lt; "D "; } };  int main(int argc, char *argv[]) {     D d;     return 0; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1

3.	<p>Что напечатает следующая программа:</p> <pre> int main() {     try     {         try         {             cout &lt;&lt; 0.5 &lt;&lt; " ";             throw 2;         }         catch ( int )         {             cout &lt;&lt; "2 ";             throw;         }         catch ( char ) { }     }     catch ( ... )     {         cout &lt;&lt; "? ";     }     return 0; } </pre>	<p>ОПК-2.3.1  ОПК-6.3.1  ОПК-6.У.1  ОПК-6.В.1</p>
4.	<p>Каким будет результат выполнения следующего кода:</p> <pre> #include &lt;stdio.h&gt;  class Parent { public:     void GetValue() { Count(); } private:     virtual void Count() { printf("%d", 1); } };  class Child : public Parent { private:     void Count() { printf("%d", 2); } };  int main() {     Parent * obj = new Child;     obj-&gt;GetValue();     return 0; } </pre>	<p>ОПК-2.3.1  ОПК-6.3.1  ОПК-6.У.1  ОПК-6.В.1</p>
5.	<p>Какое значение будет выведено в консоль в результате работы следующей программы:</p> <pre> #include &lt;iostream&gt; using namespace std;  class Base {     int* data; public:     Base(int size, int value=1)     {         data = new int(value);     }     ~Base() { delete data; }      Base &amp;operator+=(Base&amp; src)     {         *data = *data + *src.data;         return *this;     }     operator int() { return *data; } };  int main() {     Base a(2);     Base b(2, 10);     a += b;     cout &lt;&lt; b &lt;&lt; endl;     return 0; } </pre>	<p>ОПК-2.3.1  ОПК-6.3.1  ОПК-6.У.1  ОПК-6.В.1</p>



6.	<p>Что необходимо сделать, чтобы программа прошла этап компоновки (редактирования связей) и отработала корректно:</p> <pre> class Counter { public: // 1 void Count(); // 2 };  int main() { Counter obj; obj.Count(); return 0; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
7.	<p>Что напечатает следующий код:</p> <pre> class B{ public: virtual void msg () {cout &lt;&lt; "class B";} };  class D : public B{ public: virtual void msg() {cout &lt;&lt; "class D";} };  int main() { B * var = new D; var-&gt;msg(); } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
8.	<p>Какое значение будет выведено в консоль в результате работы следующей программы и почему?</p> <pre> #include &lt;iostream&gt; using namespace std;  class A { public: void someMethod(double someArg) { cout &lt;&lt; "A::someMethod()" &lt;&lt; endl; } };  class B : public A { public: void someMethod(int someArg) { cout &lt;&lt; "B::someMethod(int) someArg = "&lt;&lt; someArg &lt;&lt; endl; } };  int main(int argc, char* argv[]) { B b; b.someMethod(0.51); return 0; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
9.	<p>Что напечатает следующий код:</p> <pre> #include &lt;stdio.h&gt; class Base1 { public: virtual void Count() { printf("%d", 1); } };  class Base2 { public: virtual void Count() { printf("%d", 2); } };  class Child : public Base2, public Base1 { public: void Count() { printf("%d", 3); } };  int main() { Base2 *obj = new Child; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1

	<pre> obj-&gt;Count(); delete obj; return 0; } </pre>	
10.	<p>Что напечатает следующий код при создании экземпляра класса X:</p> <pre> #include &lt;iostream&gt; using namespace std;  class Y { public:     Y() { cout &lt;&lt; "Y"; } }; class Z { public:     Z() { cout &lt;&lt; "Z"; } }; class X : public Z { private:     Y m_objY; public:     X() { cout &lt;&lt; "X"; } };  int main() {     X x;      return 1; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
11.	<p>Что необходимо исправить в коде, чтобы он скомпилировался без ошибок:</p> <pre> class A { public:     A(void);     ~A(int); };  A::A(void) { }  void A::~~A(int a) { }  int main() {     A a; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
12.	<p>Скомпилируется ли следующий программный код:</p> <pre> class Parent { public:     ~Parent() { }     virtual void method() { } };  class Child : public Parent { public:     Child() { /* захват ресурсов */ }     ~Child() { /* освобождение ресурсов */ }     void method() { /* программный код */ } };  int main() {     Parent * obj = new Child;     // программный код     delete obj;     return 0; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
13.	<p>Скомпилируется ли следующая программа (если нет, то почему?):</p> <pre> #include &lt;iostream&gt; using namespace std; </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1

	<pre> class A { public:     virtual void printName() = 0;     virtual ~A() = 0; private:     int i; };  A::~A() {} class B : public A { public:     void printName() { cout &lt;&lt; "Class B" &lt;&lt; endl; } private:     int i_b; };  int main() {     A objA;     B objB;      return 0; } </pre>	ОПК-6.В.1
14.	<p>Что выведет на печать данный код:</p> <pre> #include &lt;iostream&gt; using namespace std;  union X {     int a;     int b; } x; struct Y {     int a;     int b; } y;  int main(int argc, char *argv[]) {     x.a = 10;     x.b = 20;      y.a = 1;     y.b = 2;     cout &lt;&lt; x.a &lt;&lt; " " &lt;&lt; x.b &lt;&lt; " " &lt;&lt; y.a &lt;&lt; " " &lt;&lt; y.b &lt;&lt; endl;      return 0; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
15.	<p>Скомпилируется ли данный код:</p> <pre> class A {     int n; public:     int k; };  int main(int argc, char *argv[]) {     A a;     a.k = 1;     a.n = 3.9;      return 0; } </pre>	ОПК-2.3.1 ОПК-6.3.1 ОПК-6.У.1 ОПК-6.В.1
16.	<p>Скомпилируется ли данный код:</p> <pre> struct A {     int n; public:     int k; }; </pre>	ОПК-6.У.1 ОПК-6.В.1

	<pre>int main(int argc, char *argv[]) {     A a;     a.k = 1;     a.n = 3.9;      return 0; }</pre>	
17.	<p>Что выведет на печать данный код:</p> <pre>#include &lt;iostream&gt; using namespace std;  void showstat(int curr) {     static int n = 1;      n += curr;      cout &lt;&lt; n &lt;&lt; " "; }  int main() {     for (int i = 0; i &lt; 3; i++)         showstat(i);     return 0; }</pre>	ОПК-6.У.1 ОПК-6.В.1
18.	<p>Что выведет на печать данный код:</p> <pre>#include &lt;iostream&gt; using namespace std;  class A { public:     static int n; };  int A::n = 0;  int main() {     A a1;     A a2;      a1.n = 10;     a2.n = 20;      cout &lt;&lt; a1.n &lt;&lt; " ";     cout &lt;&lt; a2.n &lt;&lt; endl;      return 0; }</pre>	ОПК-2.3.1 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

#### 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, приходиться 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, 4231\_Иванов\_ЛР1).

Отчет включает в себя:

- титульный лист;
- условие (с указанием номера варианта);
- полный текст (листинг) программы;
- скриншоты с результатами (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
- Введение
- Основная часть
- 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.

#### 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.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

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 являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

#### 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.

Перед выполнением лабораторных работ проводится экспресс-опрос студентов.

За выполненные в течение семестра работы студент накапливает баллы, которые будут учтены 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:

– экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

– дифференцированный зачет – это форма оценки знаний, полученных обучающимся при изучении дисциплины, при выполнении курсовых проектов, курсовых работ, научно-исследовательских работ и прохождении практик с аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

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

Экзамен или диф.зачет может проводиться путем тестирования или устного опроса по билетам.

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
