

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)

А.А. Ключарев

(инициалы, фамилия)



(подпись)

«04» июня 2020 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Объектно-ориентированное программирование»
(Наименование дисциплины)

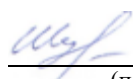
Код направления подготовки/ специальности	09.03.04
Наименование направления подготовки/ специальности	Программная инженерия
Наименование направленности	Проектирование программных систем
Форма обучения	заочная

Санкт-Петербург – 2020

Лист согласования рабочей программы дисциплины

Программу составил (а)

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

 04 июня 2020 г.
(подпись, дата)


Е.О.Шумова
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«04» июня 2020 г, протокол № 08-2019/20

Заведующий кафедрой № 43

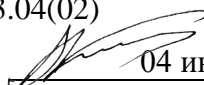
д.т.н., проф.
(уч. степень, звание)

 04 июня 2020 г.
(подпись, дата)

М.Ю. Охтилев
(инициалы, фамилия)

Ответственный за ОП ВО 09.03.04(02)

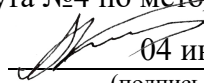
доц.,к.т.н.,доц.
(должность, уч. степень, звание)

 04 июня 2020 г.
(подпись, дата)

А.А. Ключарев
(инициалы, фамилия)

Заместитель директора института №4 по методической работе

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

 04 июня 2020 г.
(подпись, дата)

А.А. Ключарев
(инициалы, фамилия)

Аннотация

Дисциплина «Объектно-ориентированное программирование» входит в образовательную программу высшего образования 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 использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности»

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 объектно-ориентированным анализом предметных областей, объектно-ориентированным проектированием программных систем, созданием, отладкой и тестированием объектно-ориентированного программного 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 11 зачетных единиц, 396 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 использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ОПК-2.3.1 знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
Общепрофессиональные компетенции	ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	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.2 умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ ОПК-6.В.3 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:

- «Основы программирования»»,
- «Структуры и алгоритмы обработки 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
Общая трудоемкость дисциплины, ЗЕ/ (час)	11/ 396	6/ 216	5/ 180
Аудиторные занятия, всего час.	48	24	24
в том числе:			
лекции (Л), (час)	20	12	8
практические/семинарские занятия (ПЗ), (час)	8		8
лабораторные работы (ЛР), (час)	20	12	8
курсовой проект (работа) (КП, КР), (час)	*		*
экзамен, (час)	9	9	
Самостоятельная работа, всего (час)	339	183	156
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз., Дифф. Зач.	Экз.	Дифф. Зач.

Примечание: ** кандидатский экзамен

* - часы , не входящие в аудиторную нагрузку

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. Основы объектно-ориентированного программирования.	4		4		30
Раздел 2. Перегрузка операций			2		30
Раздел 3. Наследование	4		2		30
Раздел 4. Иерархия классов ввода-вывода					10

Раздел 5. Обработка исключений	1		2		33
Раздел 6. Шаблоны	1		2		50
Итого в семестре:	12		12		183
Семестр 5					
Раздел 7. Основы QT	2	4	2		56
Раздел 8. Паттерны проектирования	6	4	6		50
Выполнение курсового проекта				0	50
Итого в семестре:	8	8	8		156
Итого	20	8	20	0	339

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	Основы объектно-ориентированного программирования. 1.1. Основные понятия ООП: абстракция, инкапсуляция, наследование, полиморфизм. Классы. Синтаксис объявления. Модификаторы доступа. Сетторы и гетторы. Создание экземпляров классов, ссылки и указатели на объекты. 1.2. Конструкторы. Назначение конструкторов. Конструктор по умолчанию, копирования, с параметрами. Деструкторы. Назначение деструкторов.
3	Наследование 3.1. Одиночное наследование. Синтаксис объявления производного класса. Управление доступом в производном классе. Конструкторы и деструкторы при наследовании. 3.2. Назначение множественного наследования. Виртуальные функции и полиморфизм. Чисто виртуальные функции. Виртуальные базовые классы. Виртуальное наследование.
5	Обработка исключений 5.1. Механизм обработки исключительных ситуаций. Создание собственных исключений. Создание собственного класса исключений.
6	Шаблоны 6.1. Шаблоны функций C++. Шаблоны классов. Последовательные контейнеры. Ассоциативные контейнеры.
7	Основы QT 7.1. Обзор иерархии классов Qt. Механизм сигналов и слотов.
8	Паттерны проектирования 8.1. Общие сведения о паттернах. Смысл паттернов.

	Классификация паттернов. 8.2. Порождающие шаблоны проектирования. 8.3. Поведенческие шаблоны проектирования. Структурные 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)	№ раздела дисциплины
Семестр 5				
1	Применение паттернов проектирования	Развернутая беседа	4	8
2	Разработка интерфейса приложения	Развернутая беседа	4	7
Всего			8	

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)	№ раздела дисциплины
Семестр 4			
1.	Классы. Определение методов класса. Права доступа.	2	1
2.	Создание класса с конструкторами и деструктором	2	1
3.	Перегрузка операций через методы класса. Перегрузка с использованием дружественных функций	2	2
4.	Одиночное наследование	2	3
5.	Обработка исключительных ситуаций	2	5
6.	Использование обобщенных алгоритмов	2	6
Семестр 5			
7.	Создание приложения в QtDesigner	2	7
8.	Разработка программы с использованием шаблонов проектирования	6	8
Всего		20	

4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсового проекта: приобретение практических навыков применения моделей и методов объектно-ориентированной парадигмы, UML-моделей программ в процессах проектирования, 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)	176	100	76
Курсовое проектирование (КП, КР)	50		50
Подготовка к текущему контролю успеваемости (ТКУ)	30	20	10
Контрольные работы заочников (КРЗ)	20	20	
Подготовка к промежуточной аттестации (ПА)	63	43	20
Всего:	339	183	156

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 C++ с примерами и задачами : [учебное пособие] / А. Н. Васильев. - СПб. : Наука и техника, 2010. - 480 с.	Отдел фундаментальной литературы – 30
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., Влссидес 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

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
	Не предусмотрено

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; Задачи; Тесты.
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 5-балльная шкала	Характеристика сформированных компетенций
«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.

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
«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
1.	ООП – определение. Основные понятия ООП: Абстракция, наследование.
2.	ООП – определение. Основные понятия ООП: Инкапсуляция, полиморфизм.
3.	ООП – основные принципы. Определение класса, объекта. Их взаимосвязь между собой.
4.	C++ - Пространства имен. Объявление и способы использования.
5.	C++ - классы памяти.
6.	C++ - Универсальная инициализация. Инициализация по умолчанию базовых типов.
7.	Реализация классов в C++. Объявление и определение класса. Данные-члены и функции-члены класса.
8.	Реализация классов в C++. Класс, структура, объединение (class, struct, union) основные отличия.
9.	Реализация классов в C++. Права доступа к членам класса.
10.	Реализация классов в C++. Друзья класса. (Дружественные классы и функции).
11.	Реализация классов в C++. Перегрузка функций-членов.
12.	Реализация классов в C++. Статические и нестатические члены.
13.	Конструкторы. Конструкторы по умолчанию.
14.	Конструкторы. Инициализирующий конструктор, список инициализации и конструктор копирования.
15.	Конструкторы. Конструктор перемещения.
16.	Деструкторы. Назначение деструктора. Виртуальные деструкторы.
17.	Деструкторы. Использование деструкторов. Порядок уничтожения.
18.	Реализация классов в C++. Указатели на объекты. Указатель this.
19.	Реализация классов в C++. Указатели на члены класса.
20.	Реализация классов в C++. Шаблоны классов. Параметризация

	шаблонов классов.
21.	Реализация классов в C++. Шаблоны классов. Специализация шаблонов классов.
22.	Реализация классов в C++. Перегрузка операторов.
23.	Реализация классов в C++. Операторные функции в пространстве имен и как члены класса.
24.	Реализация классов в C++. Операторы ввода-вывода.
25.	Реализация классов в C++. Перегрузка операторов преобразования типов.
26.	Реализация классов в C++. Оператор явного (explicit) преобразования типов.
27.	Реализация наследования в C++. Одиночное наследование. Область видимости членов класса.
28.	Реализация наследования в C++. Множественное и виртуальное наследование.
29.	Реализация полиморфизма в C++. Виртуальные функции.
30.	Реализация полиморфизма в C++. Абстрактные классы. Ограничения на использование абстрактных классов.
31.	Исключения. Контролируемый блок операций. Операция генерации исключения (throw).
32.	Исключения. Реализация обработки исключительных ситуаций. Вложенные контролируемые блоки операций.
33.	Исключения. Классы стандартной библиотеки для обработки исключений.
34.	C++ - тип auto.
35.	C++ - диапазонный цикл for (...). Синтаксис, применение.
36.	C++ - Ключевое слово decltype, назначение.
37.	C++ - Перечисления с ограниченной областью видимости.
38.	Стандартная библиотека C++. Библиотека ввода-вывода. Назначение, основные классы.
39.	Стандартная библиотека C++. Контейнеры (понятие), принципы использования.
40.	Стандартная библиотека C++. Итераторы (понятие), виды.
41.	Стандартная библиотека C++. Последовательные контейнеры.
42.	Стандартная библиотека C++. Ассоциативные контейнеры.
43.	Стандартная библиотека C++. Неупорядоченные ассоциативные контейнеры.
44.	Стандартная библиотека C++. Работа со строками символов.
45.	Стандартная библиотека C++. Поток (thread).
46.	Шаблоны проектирования (Design patterns). Виды.

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
1	Основные понятия объектно-ориентированного программирования.
2	Определение класса. Создание объектов класса. Привести пример.
3	Модификаторы доступа к членам класса. Пример использования.
4	Статические и нестатические члены класса. Пример использования.
5	Гетторы и сетторы. Назначение. Пример использования

6	Конструкторы и деструкторы класса. Привести пример.
7	Класс. Структура. Объединение. Основные отличия. Привести пример.
8	Указатели на объекты. Указатель this. Пример.
9	Дружественные функции. Пример использования.
10	Перегрузка операций. Пример использования.

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 отсутствии).</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 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</i>

	<i>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	<i>Разработка системы классов для обеспечения работы больницы (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
1.	<p>Что выведет следующий код при создании экземпляра класса D?</p> <pre> struct A { A(){ cout << "A "; } }; struct B : public A { B() { cout << "B "; } }; struct C : public A { C() { cout << "C "; } }; struct D : B, C { D() { cout << "D "; } }; int main(int argc, char *argv[]) { D d; return 0; } </pre>
2.	<pre> struct A { ~A(){ cout << "A "; } }; struct B : public A { ~B() { cout << "B "; } }; struct C : public A { ~C() { cout << "C "; } }; struct D : B, C { ~D() { cout << "D "; } }; int main(int argc, char *argv[]) { </pre>

	<pre> D d; return 0; } </pre>
3.	<p>Что напечатает следующая программа:</p> <pre> int main() { try { try { cout << 0.5 << " "; throw 2; } catch (int) { cout << "2 "; throw; } catch (char) { } } catch (...) { cout << "? "; } return 0; } </pre>
4.	<p>Каким будет результат выполнения следующего кода:</p> <pre> #include <stdio.h> 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->GetValue(); return 0; } </pre>
5.	<p>Какое значение будет выведено в консоль в результате работы следующей программы:</p> <pre> #include <iostream> using namespace std; class Base { int* data; public: Base(int size, int value=1) { data = new int(value); } ~Base() { delete data; } } </pre>

	<pre> Base &operator+=(Base& src) { *data = *data + *src.data; return *this; } operator int() { return *data; } }; int main() { Base a(2); Base b(2, 10); a += b; cout << b << endl; return 0; } </pre>
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>
7.	<p>Что напечатает следующий код:</p> <pre> class B{ public: virtual void msg () {cout << "class B";} }; class D : public B{ public: virtual void msg() {cout << "class D";} }; int main() { B * var = new D; var->msg(); } </pre>
8.	<p>Какое значение будет выведено в консоль в результате работы следующей программы и почему?</p> <pre> #include <iostream> using namespace std; class A { public: void someMethod(double someArg) { cout << "A::someMethod()" << endl; } }; class B : public A { public: void someMethod(int someArg) { cout << "B::someMethod(int) someArg = "<< someArg << endl; } }; </pre>

	<pre>int main(int argc, char* argv[]) { B b; b.someMethod(0.51); return 0; }</pre>
9.	<p>Что напечатает следующий код:</p> <pre>#include <stdio.h> 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; obj->Count(); delete obj; return 0; }</pre>
10.	<p>Что напечатает следующий код при создании экземпляра класса X:</p> <pre>#include <iostream> using namespace std; class Y { public: Y() { cout << "Y"; } }; class Z { public: Z() { cout << "Z"; } }; class X : public Z { private: Y m_objY; public: X() { cout << "X"; } }; int main() { X x; return 1; }</pre>

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>
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>
13.	<p>Скомпилируется ли следующая программа (если нет, то почему?):</p> <pre> #include <iostream> using namespace std; class A { public: virtual void printName() = 0; virtual ~A() = 0; private: int i; }; A::~~A() {} class B : public A { public: void printName() { cout << "Class B" << endl; } private: </pre>

	<pre> int i_b; }; int main() { A objA; B objB; return 0; } </pre>
14.	<p>Что выведет на печать данный код:</p> <pre> #include <iostream> 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 << x.a << " " << x.b << " " << y.a << " " << y.b << endl; return 0; } </pre>
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>
16.	<p>Скомпилируется ли данный код:</p> <pre> struct A { int n; public: int k; }; </pre>

	<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 <iostream> using namespace std; void showstat(int curr) { static int n = 1; n += curr; cout << n << " "; } int main() { for (int i = 0; i < 3; i++) showstat(i); return 0; }</pre>
18.	<p>Что выведет на печать данный код:</p> <pre>#include <iostream> 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 << a1.n << " "; cout << a2.n << endl; return 0; }</pre>

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

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
1	Дайте определение конструктора. Каково назначение конструктора? Перечислите отличия конструктора от метода.

2	Дайте определение дружественной функции. Как объявляется дружественная функция? А как определяется?
3	Сколько конструкторов может быть в классе? Допускается ли перегрузка конструкторов? Какие виды конструкторов создаются по умолчанию?
4	Может ли конструктор быть приватным? Какие последствия влечет за собой объявление конструктора приватным?
5	Приведите несколько случаев, когда конструктор вызывается неявно.
6	Каким образом разрешается инициализировать константные поля в классе?
7	В каком порядке инициализируются поля в классе? Совпадает ли этот порядок с порядком перечисления инициализаторов в списке инициализации конструктора?
8	Для чего нужны статические поля в классе? Как они определяются?
9	Что такое деструктор? Может ли деструктор иметь параметры?
10	Допускается ли перегрузка деструкторов?
11	Каковы функции контролируемого блока?
12	Какого типа может быть исключение?
13	Объясните, каким образом преодолеть ограничение на передачу единственного параметра в блок обработки.
14	Напишите конструкцию, которая позволяет перехватить любое исключение.
15	Могут ли контролируемые блоки быть вложенными?
16	Какие виды наследования возможны в C++?
17	Чем отличается модификатор доступа <code>protected</code> от модификаторов <code>private</code> и <code>public</code> ?
18	Чем открытое наследование отличается от закрытого и защищенного?
19	Сформулируйте правила написания конструкторов в производном классе.
20	Каков порядок вызова конструкторов? А деструкторов?
21	Если имя нового поля совпадает с именем унаследованного, то каким образом разрешить конфликт имен?
22	Что происходит, если имя метода-наследника совпадает с именем базового метода?
23	Объясните, зачем нужны виртуальные функции.
24	Дайте определение абстрактного класса.
25	Наследуются ли чистые виртуальные функции?
26	Можно ли объявить деструктор чисто виртуальным?
27	Сколько места в классе занимают статические поля?
28	Чем отличается статический метод от обычного?
29	Сформулируйте основную проблему множественного наследования.
30	Что такое виртуальное наследование? Каковы его преимущества и недостатки по сравнению с обычным наследованием?

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 *(если предусмотрено учебным планом по данной дисциплине)*.

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 *(если предусмотрено учебным планом по данной дисциплине)*

Практическое занятие является одной из основных форм организации учебного процесса, заключающаяся в выполнении обучающимися под руководством преподавателя комплекса учебных заданий с целью усвоения научно-теоретических основ учебной дисциплины, приобретения умений и навыков, опыта творческой деятельности.

Целью практического занятия для обучающегося является привитие обучающимся умений и навыков практической деятельности по изучаемой дисциплине.

Планируемые результаты при освоении обучающимися практических занятий:

- закрепление, углубление, расширение и детализация знаний при решении конкретных задач;
- развитие познавательных способностей, самостоятельности мышления, творческой активности;
- овладение новыми методами и методиками изучения конкретной учебной дисциплины;
- выработка способности логического осмысления полученных знаний для выполнения заданий;
- обеспечение рационального сочетания коллективной и индивидуальной форм обучения.

Требования к проведению практических занятий

Для проведения практических занятий необходим компьютерный класс с установленным программным обеспечением, перечисленным в таблице 10.

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 фамилию студента, № работы, № группы (например, Иванов_ЛР1_4931).

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:

Структура пояснительной записки курсового проекта/ работы

- Титульный лист
- Задание на курсовое проектирование

- Содержание
- Введение
- Основная часть (постановка зада
- 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).

Если методические указания по прохождению самостоятельной работы имеются в изданном виде, в виде электронных ресурсов библиотеки ГУАП, системы LMS, кафедры и т.д., необходимо дать на них ссылку или привести URL адрес.

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».

– зачет – это форма оценки знаний, полученных обучающимся в ходе изучения учебной дисциплины в целом или промежуточная (по окончании семестра) оценка знаний обучающимся по отдельным разделам дисциплины с аттестационной оценкой «зачтено» или «не зачтено».

– 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