



Федеральное государственное бюджетное образовательное учреждение  
высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ

Кафедра Информационных технологий

УТВЕРЖДАЮ

Начальник учебно-методического управления

С.В. Михайлов

«29» июня 2021г.

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

Объектно-ориентированное программирование

направление подготовки/специальность 01.03.02 Прикладная математика и информатика

направленность (профиль)/специализация образовательной программы Прикладная математика и информатика

Форма обучения очная

Санкт-Петербург, 2021

## 1. Цели и задачи освоения дисциплины (модуля)

Целями освоения дисциплины являются получение знаний, умений и навыков в области разработки объектно-ориентированных программ на языке С++ для решения прикладных задач в различных сферах деятельности на базе:

- системного изложения основ алгоритмического языка С++, его возможностей по разработке объектно-ориентированных программ и Windows-приложений;
- ознакомления студентов с технологиями и инструментальными средствами разработки программного обеспечения, основными структурами данных и методами работы с ними, в том числе с использованием стандартной библиотеки классов языка С++.

Задачами освоения дисциплины являются обучение студентов навыкам постановки задачи, разработки алгоритмов, выбора структуры данных, составления как процедурно-ориентированных, так и объектно-ориентированных программ на языке С++ для решения широкого круга практических задач в инженерных и экономических расчетах, обработки текстовой, графической и другой информации.

## 2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, обеспечивающие достижение планируемых результатов освоения ОПОП
ПКС-1 Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	ПКС-1.1 знает основные языки и концепции программирования	
ПКС-1 Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	ПКС-1.2 умеет работать с современными средствами разработки программного обеспечения (ПО)	
ПКС-1 Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	ПКС-1.3 имеет навыки разработки ПО с использованием современных инструментальных средств	

## 3. Указание места дисциплины (модуля) в структуре образовательной программы

Данная дисциплина (модуль) включена в Блок «Дисциплины, модули» Б1.В.04 основной профессиональной образовательной программы 01.03.02 Прикладная математика и информатика и относится к части, формируемой участниками образовательных отношений учебного плана.

№ п/п	Предшествующие дисциплины	Код и наименование индикатора достижения компетенции
-------	---------------------------	--

1	Алгоритмы и алгоритмические языки	УК-1.1, ОПК-2.2, ОПК-2.3, ОПК-4.2, ОПК-5.1, ОПК-5.2
---	-----------------------------------	---

Алгоритмы и алгоритмические языки

знать:

- основные понятия информатики;
- основы разработки алгоритмов;
- основные типы данных;
- основные структуры данных;
- основные механизмы обработки информации в памяти компьютера.

уметь:

- работать на персональном компьютере;
- пользоваться операционной системой;
- использовать возможности вычислительной техники и программного обеспечения.

владеть:

- первичными навыками и основными методами решения математических задач;
- навыками работы с учебной литературой;
- основными приёмами работы на компьютере с прикладным программным обеспечением.

№ п/п	Последующие дисциплины	Код и наименование индикатора достижения компетенции
1	Программирование для Интернет	ОПК-4.1, ОПК-4.2, ОПК-4.3

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

Вид учебной работы	Всего часов	Из них часы на практическую подготовку	Семестр		
			3	4	5
<b>Контактная работа</b>	176		64	48	64
Лекционные занятия (Лек)	16	0		16	
Лабораторные занятия (Лаб)	160	0	64	32	64
<b>Иная контактная работа, в том числе:</b>	1,4		0,25	0,65	0,5
консультации по курсовой работе (проекту), контрольным работам (РГР)	1,4			0,4	1
контактная работа на аттестацию (сдача зачета, зачета с оценкой; защита курсовой работы (проекта); сдача контрольных работ (РГР))	0,65			0,4	0,25
контактная работа на аттестацию в сессию (консультация перед экзаменом и сдача	0,75		0,25	0,25	0,25
<b>Часы на контроль</b>	62,25		8,75	26,75	26,75
<b>Самостоятельная работа (СР)</b>	154,95		35	68,2	51,75
<b>Общая трудоемкость дисциплины (модуля)</b>					
<b>часы:</b>	396		108	144	144
<b>зачетные единицы:</b>	11		3	4	4

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

5.1. Тематический план дисциплины (модуля)



2.1.	Зачет с оценкой	3							9	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.	3 раздел. Основы объектно-ориентированного программирования									
3.1.	Объекты и классы. Данные класса. Методы класса. Передача объектов в функции.	4	1			2		8	11	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.2.	Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты.	4	1			2		8,2	11,2	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.3.	Массивы как члены классов. Строки как члены классов.	4	2			4		8	14	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.4.	Указатели на объекты. Использование классов для создания структур для хранения данных.	4	2			4		8	14	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.5.	Перегрузка операций.	4	2			4		8	14	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.6.	Наследование. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование.	4	2			4		8	14	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.7.	Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции.	4	2			4		8	14	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.8.	Потоки и файлы. Поточковые классы. Поточковый ввод/вывод дисковых файлов.	4	2			4		6	12	ПКС-1.1, ПКС-1.2, ПКС-1.3
3.9.	Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения.	4	2			4		6	12	ПКС-1.1, ПКС-1.2, ПКС-1.3
4.	4 раздел. Иная контактная работа									
4.1.	Иная контактная работа	4							0,8	ПКС-1.1, ПКС-1.2, ПКС-1.3
5.	5 раздел. Контроль									
5.1.	Экзамен	4							27	ПКС-1.1, ПКС-1.2, ПКС-1.3

6.	6 раздел. Создание приложений с использованием стандартных библиотек										
6.1.	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL.	5				4		10	14	ПКС-1.1, ПКС-1.2, ПКС-1.3	
6.2.	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров.	5				12		8	20	ПКС-1.1, ПКС-1.2, ПКС-1.3	
6.3.	Итераторы STL. Адаптеры итераторов. Потокотворные итераторы. Ассоциативные контейнеры.	5				12		7	19	ПКС-1.1, ПКС-1.2, ПКС-1.3	
6.4.	Основы разработки Windows-приложений в среде Microsoft Visual Studio. Графическая библиотека MFC.	5				8		4	12	ПКС-1.1, ПКС-1.2, ПКС-1.3	
6.5.	Основные этапы создания Windows-приложения с использованием MFC. Создание простейших элементов управления.	5				8		5,75	13,75	ПКС-1.1, ПКС-1.2, ПКС-1.3	
6.6.	Повышение функциональности Windows-приложения с использованием MFC.	5				8		6	14	ПКС-1.1, ПКС-1.2, ПКС-1.3	
6.7.	Разработка объектно-ориентированного программного обеспечения. Эволюция процесса создания программного обеспечения.	5				12		11	23	ПКС-1.1, ПКС-1.2, ПКС-1.3	
7.	7 раздел. Иная контактная работа										
7.1.	Иная контактная работа	5							1,25	ПКС-1.1, ПКС-1.2, ПКС-1.3	
8.	8 раздел. Контроль										
8.1.	Экзамен	5							27	ПКС-1.1, ПКС-1.2, ПКС-1.3	

### 5.1. Лекции

№ п/п	Наименование раздела и темы лекций	Наименование и краткое содержание лекций
11	Объекты и классы. Данные класса. Методы класса. Передача объектов в функции.	Объекты и классы. Данные класса. Методы класса. Передача объектов в функции. Объекты и классы. Данные класса. Методы класса. Объекты программы и объекты реального мира. Объекты в качестве аргументов функций. Объекты, возвращаемые функцией.

12	Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты.	Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты. Перегруженные конструкторы. Определение методов класса вне класса. Аргументы и объекты. Структуры и классы. Статические данные класса. Константные методы. Константные аргументы методов. Константные объекты.
13	Массивы как члены классов. Строки как члены классов.	Массивы как члены классов. Строки как члены классов. Массивы объектов. Строки как члены классов. Определенные пользователем типы строк. Стандартный класс string языка C++.
14	Указатели на объекты. Использование классов для создания структур для хранения данных.	Указатели на объекты. Использование классов для создания структур для хранения данных. Указатели на объекты, использование классов для создания структур для хранения данных. Стек. Очередь. Связный список.
15	Перегрузка операций.	Перегрузка операций. Перегрузка унарных операций. Перегрузка бинарных операций. Множественная перегрузка. Объединение строк, сравнение строк. Преобразование типов. Преобразование объектов.
16	Наследование. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование.	Наследование. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование. Наследование. Базовый и производный классы. Конструкторы производного класса. Базовые функции класса. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование. Роль наследования при разработке объектно-ориентированных программ.
17	Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции.	Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции. Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Дружественные функции и перегрузка операций. Статические функции.
18	Потоки и файлы. Потокосовые классы. Потокосовый ввод/вывод дисковых файлов.	Потоки и файлы. Потокосовые классы. Потокосовый ввод/вывод дисковых файлов. Потокосовый ввод/вывод дисковых файлов. Указатели файлов. Файловый ввод/вывод с помощью методов. Многофайловые программы. Межфайловое взаимодействие.
19	Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения.	Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения. Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения.

## 5.2. Лабораторные работы

№ п/п	Наименование раздела и темы лабораторных работ	Наименование и содержание лабораторных работ
1	Введение. Общая характеристика и сравнительный анализ	Введение. Общая характеристика и сравнительный анализ объектно-ориентированного и структурного программирования. Введение. Общая характеристика ООП: данные и функции, объекты

	объектно-ориентированного и структурного программирования.	и классы, наследование и полиморфизм, пользовательские типы данных. Сравнительный анализ объектно-ориентированного и структурного программирования. Основы работы в среде программирования Microsoft Visual Studio.
2	Структура программы. Типы данных. Библиотечные файлы. Форматирование ввода/вывода.	Структура программы. Типы данных. Библиотечные файлы. Форматирование ввода/вывода. Структура программы. Типы данных, преобразования типов. Директивы и заголовочные файлы. Библиотечные файлы. Объекты для работы со стандартными потоками ввода-вывода. Манипуляторы и флаги форматирования.
3	Организация вычислений. Циклы. Ветвления.	Организация вычислений. Циклы. Ветвления. Организация вычислений. Виды и приоритеты операций в C++. Циклы. Ветвления.
4	Пользовательские типы данных. Структуры и перечисления. Структуры и классы.	Пользовательские типы данных. Структуры и перечисления. Структуры и классы. Пользовательские типы данных. Преобразование пользовательских типов данных. Структуры и перечисления. Структуры и классы: сходство и различия.
5	Функции и переменные. Передача аргументов в функцию. Возврат функцией значения.	Функции и переменные. Передача аргументов в функцию. Возврат функцией значения. Функции и переменные. Передача аргументов в функцию: передача по значению, передача по ссылке, передача структурных переменных. Значение, возвращаемое функцией. Аргументы по умолчанию, константные аргументы.
6	Функции и переменные. Пользовательские и библиотечные функции. Перегруженные функции. Встраиваемые функции. Область видимости и класс памяти переменных.	Функции и переменные. Пользовательские и библиотечные функции. Перегруженные функции. Встраиваемые функции. Область видимости и класс памяти переменных. Функции и переменные. Пользовательские и библиотечные функции. Перегруженные функции. Встраиваемые функции. Область видимости и класс памяти переменных в C++. Глобальные, локальные и статические переменные.
7	Массивы и строки. Виды массивов. Передача массивов в функцию. Виды строк в C++. Методы и объекты для работы со строками.	Массивы и строки. Виды массивов. Передача массивов в функцию. Виды строк в C++. Методы и объекты для работы со строками. Массивы и строки. Массивы как класс данных. Одномерные и многомерные массивы. Массивы структур. Передача массивов в функцию. Виды строк в C++. Методы и объекты для работы со строками. Массивы строк.
8	Указатели, адреса и выделение памяти. Адресная арифметика. Динамические массивы.	Указатели, адреса и выделение памяти. Адресная арифметика. Динамические массивы. Указатели, адреса и выделение памяти. Адресная арифметика. Операция получения адреса. Указатели и массивы. Указатели и функции. Указатели на строки. Выделение памяти с помощью операции new. Освобождение памяти с помощью операции delete. Динамические массивы.
9	Взаимодействие C++ со средствами операционной системы.	Взаимодействие C++ со средствами операционной системы. Возможности C++ по взаимодействию со средствами операционной системы. Расширенные функции по выводу текстовой информации: перемещение по экрану, управление цветом фона и текста, флаги яркости.



11	Объекты и классы. Данные класса. Методы класса. Передача объектов в функции.	Объекты и классы. Данные класса. Методы класса. Передача объектов в функции. Объекты и классы. Данные класса. Методы класса.
12	Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты.	Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты. Конструкторы и деструкторы.
13	Массивы как члены классов. Строки как члены классов.	Массивы как члены классов. Строки как члены классов. Массивы объектов.
14	Указатели на объекты. Использование классов для создания структур для хранения данных.	Указатели на объекты. Использование классов для создания структур для хранения данных. Указатели на объекты.
15	Перегрузка операций.	Перегрузка операций. Перегрузка операций.
16	Наследование. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование.	Наследование. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование. Наследование.
17	Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции.	Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции. Виртуальные функции. Дружественные функции.
18	Потоки и файлы. Потокосые классы. Потокосый ввод/вывод дисковсх файлов.	Потоки и файлы. Потокосые классы. Потокосый ввод/вывод дисковсх файлов. Потокосый ввод/вывод данных разного типа.
19	Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения.	Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения. Шаблоны и исключения.
22	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL.	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL. Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL. Методы STL. Функциональные объекты. Пользовательские функции в роли функциональных объектов.
23	Контейнеры STL.	Контейнеры STL. Виды контейнеров. Последовательные контейнеры.

	Виды контейнеров. Последовательные контейнеры. Методы контейнеров.	Методы контейнеров. Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Векторы. Списки. Очереди с двусторонним доступом. Методы контейнеров.
24	Итераторы STL. Адаптеры итераторов. Поточные итераторы. Ассоциативные контейнеры.	Итераторы STL. Адаптеры итераторов. Поточные итераторы. Ассоциативные контейнеры. Итераторы STL. Специализированные итераторы. Адаптеры итераторов. Поточные итераторы. Ассоциативные контейнеры. Множества и мультимножества. Отображения и мультиотображения. Ассоциативные массивы.
25	Основы разработки Windows-приложений в среде Microsoft Visual Studio. Графическая библиотека MFC.	Основы разработки Windows-приложений в среде Microsoft Visual Studio. Графическая библиотека MFC. Основы разработки Windows-приложений в среде Microsoft Visual Studio. Визуальное проектирование. Событийное программирование. Графическая библиотека классов Microsoft Foundation Classes (MFC). Кроссплатформенные графические библиотеки.
26	Основные этапы создания Windows-приложения с использованием MFC. Создание простейших элементов управления.	Основные этапы создания Windows-приложения с использованием MFC. Создание простейших элементов управления. Основные этапы создания Windows-приложения с использованием MFC. Создание простейших элементов управления. Кнопки, поля редактирования, текстовые поля. Переменные для полей. Связывание элементов управления с событиями.
27	Повышение функциональности Windows-приложения с использованием MFC.	Повышение функциональности Windows-приложения с использованием MFC. Повышение функциональности Windows-приложения с использованием MFC. Элементы управления списки, переключатели, флажки. Основные методы MFC. Создание меню.
28	Разработка объектно-ориентированного программного обеспечения. Эволюция процесса создания программного обеспечения.	Разработка объектно-ориентированного программного обеспечения. Эволюция процесса создания программного обеспечения. Разработка объектно-ориентированного программного обеспечения. Эволюция процесса создания программного обеспечения. Моделирование вариантов использования. Пред-метная область программирования. От вариантов использования к классам. Написание кода. Взаимодействие с программой.

### 5.3. Самостоятельная работа обучающихся

№ п/п	Наименование раздела дисциплины и темы	Содержание самостоятельной работы
1	Введение. Общая характеристика и сравнительный анализ объектно-ориентированного и структурного программирования.	Введение. Общая характеристика и сравнительный анализ объектно-ориентированного и структурного программирования. Подготовка к лабораторной работе.
2	Структура программы. Типы данных. Библиотечные файлы. Форматирование ввода/вывода.	Структура программы. Типы данных. Библиотечные файлы. Форматирование ввода/вывода. Подготовка к лабораторной работе.
3	Организация	Организация вычислений. Циклы. Ветвления.

	вычислений. Циклы. Ветвления.	Подготовка к лабораторной работе.
4	Пользовательские типы данных. Структуры и перечисления. Структуры и классы.	Пользовательские типы данных. Структуры и перечисления. Структуры и классы. Подготовка к лабораторной работе.
5	Функции и переменные. Передача аргументов в функцию. Возврат функцией значения.	Функции и переменные. Передача аргументов в функцию. Возврат функцией значения. Подготовка к лабораторной работе.
6	Функции и переменные. Пользовательские и библиотечные функции. Перегруженные функции. Встраиваемые функции. Область видимости и класс памяти переменных.	Функции и переменные. Пользовательские и библиотечные функции. Перегруженные функции. Встраиваемые функции. Область видимости и класс памяти переменных. Подготовка к лабораторной работе.
7	Массивы и строки. Виды массивов. Передача массивов в функцию. Виды строк в C++. Методы и объекты для работы со строками.	Массивы и строки. Виды массивов. Передача массивов в функцию. Виды строк в C++. Методы и объекты для работы со строками. Подготовка к лабораторной работе.
8	Указатели, адреса и выделение памяти. Адресная арифметика. Динамические массивы.	Указатели, адреса и выделение памяти. Адресная арифметика. Динамические массивы. Подготовка к лабораторной работе.
9	Взаимодействие C++ со средствами операционной системы.	Взаимодействие C++ со средствами операционной системы. Подготовка к лабораторной работе.
11	Объекты и классы. Данные класса. Методы класса. Передача объектов в функции.	Объекты и классы. Данные класса. Методы класса. Передача объектов в функции. Подготовка к лабораторной работе.
12	Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты.	Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты. Подготовка к лабораторной работе.
13	Массивы как члены классов. Строки как члены классов.	Массивы как члены классов. Строки как члены классов. Подготовка к лабораторной работе.
14	Указатели на объекты.	Указатели на объекты. Использование классов для создания структур

	Использование классов для создания структур для хранения данных.	для хранения данных. Подготовка к лабораторной работе. Подготовка к тесту №1.
15	Перегрузка операций.	Перегрузка операций. Подготовка к лабораторной работе.
16	Наследование. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование.	Наследование. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование. Подготовка к лабораторной работе. Подготовка к тесту №2.
17	Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции.	Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции. Подготовка к лабораторной работе.
18	Потоки и файлы. Потокосые классы. Потокосый ввод/вывод дисковсх файлов.	Потоки и файлы. Потокосые классы. Потокосый ввод/вывод дисковсх файлов. Подготовка к лабораторной работе.
19	Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения.	Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения. Подготовка к лабораторной работе. Подготовка к тесту №3.
22	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL.	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL. Подготовка к практическим занятиям.
23	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров.	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров. Подготовка к практическим занятиям.
24	Итераторы STL. Адаптеры итераторов. Потокосые итераторы. Ассоциативные контейнеры.	Итераторы STL. Адаптеры итераторов. Потокосые итераторы. Ассоциативные контейнеры. Подготовка к практическим занятиям. Подготовка к тесту №4.
25	Основы разработки Windows-приложений в среде Microsoft Visual Studio. Графическая библиотека MFC.	Основы разработки Windows-приложений в среде Microsoft Visual Studio. Графическая библиотека MFC. Подготовка к практическим занятиям.
26	Основные этапы создания Windows-	Основные этапы создания Windows-приложения с использованием

	приложения с использованием MFC. Создание простейших элементов управления.	MFC. Создание простейших элементов управления. Подготовка к практическим занятиям.
27	Повышение функциональности Windows-приложения с использованием MFC.	Повышение функциональности Windows-приложения с использованием MFC. Подготовка к практическим занятиям.
28	Разработка объектно-ориентированного программного обеспечения. Эволюция процесса создания программного обеспечения.	Разработка объектно-ориентированного программного обеспечения. Эволюция процесса создания программного обеспечения. Выполнение эссе.

## 6. Методические материалы для самостоятельной работы обучающихся по дисциплине (модулю)

Программой дисциплины предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал, и лабораторных занятий, предполагающих закрепление изученного материала и формирование у обучающихся необходимых знаний, умений и навыков. Кроме того, важнейшим этапом изучения дисциплины является самостоятельная работа обучающихся с использованием всех средств и возможностей современных образовательных технологий.

В объем самостоятельной работы по дисциплине включается следующее:

- изучение теоретических вопросов по всем темам дисциплины;
- выполнение курсовой работы;
- подготовка к лабораторным занятиям;
- подготовка к текущему контролю успеваемости студентов;
- подготовка к зачету с оценкой и экзамену.

Залогом успешного освоения дисциплины является обязательное посещение лекционных и лабораторных занятий, так как пропуск одного (тем более, нескольких) занятий может осложнить освоение разделов курса. На лабораторных занятиях материал, изложенный на лекциях, закрепляется при выполнении лабораторных заданий.

Приступая к изучению дисциплины, необходимо в первую очередь ознакомиться с содержанием РПД, а также методическими указаниями по организации самостоятельной работы.

При подготовке к лекционным занятиям студенту необходимо:

- ознакомиться с соответствующей темой занятия;
- осмыслить круг изучаемых вопросов и логику их рассмотрения;
- изучить рекомендуемую рабочей программой литературу по данной теме.

При подготовке к лабораторным занятиям и в рамках самостоятельной работы по изучению дисциплины обучающимся необходимо:

- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;
- при самостоятельном изучении теоретической темы сделать конспект, используя рекомендованные в РПД источники;
- выполнить лабораторные задания в рамках изучаемой темы;
- ответить на контрольные вопросы по теме, используя материалы ФОС;
- подготовиться к проверочной работе, предусмотренной в контрольных точках;
- подготовиться к промежуточной аттестации.

Итогом изучения дисциплины является зачет с оценкой экзамен. Зачет с оценкой и Экзамен проводится по расписанию. Форма проведения занятия может быть устная, письменная и в электронном виде. Студенты, не прошедшие аттестацию, должны ликвидировать задолженность в установленном порядке.

## 7. Оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине (модулю)

### 7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

№ п/п	Контролируемые разделы дисциплины (модуля)	Код и наименование индикатора контролируемой компетенции	Вид оценочного средства
1	Введение. Общая характеристика и сравнительный анализ объектно-ориентированного и структурного программирования.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №1, программы для зачета, вопросы для экзамена.
2	Структура программы. Типы данных. Библиотечные файлы. Форматирование ввода/вывода.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №1, программы для зачета, вопросы для экзамена.
3	Организация вычислений. Циклы. Ветвления.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №2, программы для зачета, вопросы для экзамена.

4	Пользовательские типы данных. Структуры и перечисления. Структуры и классы.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №2, программы для зачета, вопросы для экзамена.
5	Функции и переменные. Передача аргументов в функцию. Возврат функцией значения.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №2, программы для зачета, вопросы для экзамена.
6	Функции и переменные. Пользовательские и библиотечные функции. Перегруженные функции. Встраиваемые функции. Область видимости и класс памяти переменных.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №2, программы для зачета, вопросы для экзамена.
7	Массивы и строки. Виды массивов. Передача массивов в функцию. Виды строк в C++. Методы и объекты для работы со строками.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №3, программы для зачета, вопросы для экзамена.
8	Указатели, адреса и выделение памяти. Адресная арифметика. Динамические массивы.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №3, программы для зачета, вопросы для экзамена.
9	Взаимодействие C++ со средствами операционной системы.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №3, программы для зачета, вопросы для экзамена.
10	Зачет с оценкой	ПКС-1.1, ПКС-1.2, ПКС-1.3	
11	Объекты и классы. Данные класса. Методы класса. Передача объектов в функции.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №4, программы для зачета, вопросы для экзамена.
12	Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №4, программы для зачета, вопросы для экзамена.
13	Массивы как члены классов. Строки как члены классов.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №4, программы для зачета, вопросы для экзамена.
14	Указатели на объекты. Использование классов для создания структур для хранения данных.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №4, программы для зачета, вопросы для экзамена.
15	Перегрузка операций.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №5, программы для зачета, вопросы для экзамена.
16	Наследование. Иерархия классов. Общее и частное наследование. Уровни наследования. Множественное наследование.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №5, программы для зачета, вопросы для экзамена.
17	Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №6, программы для зачета, вопросы для экзамена.
18	Потоки и файлы. Потокосые классы. Потокосый ввод/вывод дисковых файлов.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №6, программы для зачета, вопросы для экзамена.
19	Шаблоны и исключения. Шаблоны функций. Шаблоны классов. Исключения.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №6, программы для зачета, вопросы для экзамена.
20	Иная контактная работа	ПКС-1.1, ПКС-1.2, ПКС-1.3	

21	Экзамен	ПКС-1.1, ПКС-1.2, ПКС-1.3	
22	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №7, программы для экзамена, вопросы для экзамена.
23	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №7, программы для экзамена, вопросы для экзамена.
24	Итераторы STL. Адаптеры итераторов. Поточковые итераторы. Ассоциативные контейнеры.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №7, программы для экзамена, вопросы для экзамена.
25	Основы разработки Windows-приложений в среде Microsoft Visual Studio. Графическая библиотека MFC.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №8, программы для экзамена, вопросы для экзамена.
26	Основные этапы создания Windows-приложения с использованием MFC. Создание простейших элементов управления.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №8, программы для экзамена, вопросы для экзамена.
27	Повышение функциональности Windows-приложения с использованием MFC.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Тест №8, программы для экзамена, вопросы для экзамена.
28	Разработка объектно-ориентированного программного обеспечения. Эволюция процесса создания программного обеспечения.	ПКС-1.1, ПКС-1.2, ПКС-1.3	Эссе, программы для экзамена, вопросы для экзамена.
29	Иная контактная работа	ПКС-1.1, ПКС-1.2, ПКС-1.3	
30	Экзамен	ПКС-1.1, ПКС-1.2, ПКС-1.3	

7.2. Типовые контрольные задания или иные материалы текущего контроля успеваемости, необходимые для оценки знаний, умений и навыков и (или) опыта профессиональной деятельности, характеризующих этапы формирования компетенций в процессе освоения дисциплины

Типовые контрольные задания и иные материалы текущего контроля успеваемости размещены по адресу ЭИОС Moodle (<https://moodle.spbgasu.ru/course/view.php?id=456>)

7.3. Система оценивания результатов обучения по дисциплине (модулю) при проведении текущего контроля успеваемости



<p>Оценка «отлично» (зачтено)</p>	<p>знания:</p> <ul style="list-style-type: none"> <li>- систематизированные, глубокие и полные знания по всем разделам дисциплины, а также по основным вопросам, выходящим за пределы учебной программы;</li> <li>- точное использование научной терминологии, систематически грамотное и логически правильное изложение ответа на вопросы;</li> <li>- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю)</li> </ul> <p>умения:</p> <ul style="list-style-type: none"> <li>- умеет ориентироваться в теориях, концепциях и направлениях дисциплины и давать им критическую оценку, используя научные достижения других дисциплин</li> </ul> <p>навыки:</p> <ul style="list-style-type: none"> <li>- высокий уровень сформированности заявленных в рабочей программе компетенций;</li> <li>- владеет навыками самостоятельно и творчески решать сложные проблемы и нестандартные ситуации;</li> <li>- применяет теоретические знания для выбора методики выполнения заданий;</li> <li>- грамотно обосновывает ход решения задач;</li> <li>- безупречно владеет инструментарием учебной дисциплины, умение его эффективно использовать в постановке научных и практических задач;</li> <li>- творческая самостоятельная работа на практических/семинарских/лабораторных занятиях, активно участвует в групповых обсуждениях, высокий уровень культуры исполнения заданий</li> </ul>
<p>Оценка «хорошо» (зачтено)</p>	<p>знания:</p> <ul style="list-style-type: none"> <li>- достаточно полные и систематизированные знания по дисциплине;</li> <li>- усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю)</li> </ul> <p>умения:</p> <ul style="list-style-type: none"> <li>- умеет ориентироваться в основных теориях, концепциях и направлениях дисциплины и давать им критическую оценку;</li> <li>- использует научную терминологию, лингвистически и логически правильно излагает ответы на вопросы, умеет делать обоснованные выводы;</li> <li>- владеет инструментарием по дисциплине, умение его использовать в постановке и решении научных и профессиональных задач</li> </ul> <p>навыки:</p> <ul style="list-style-type: none"> <li>- самостоятельная работа на практических занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий;</li> <li>- средний уровень сформированности заявленных в рабочей программе компетенций;</li> <li>- без затруднений выбирает стандартную методику выполнения заданий;</li> <li>- обосновывает ход решения задач без затруднений</li> </ul>

<p>Оценка «удовлетворительно» (зачтено)</p>	<p>знания: - достаточный минимальный объем знаний по дисциплине; - усвоение основной литературы, рекомендованной рабочей программой; - использование научной терминологии, стилистическое и логическое изложение ответа на вопросы, умение делать выводы без существенных ошибок умения: - умеет ориентироваться в основных теориях, концепциях и направлениях по дисциплине и давать им оценку; - владеет инструментарием учебной дисциплины, умение его использовать в решении типовых задач; - умеет под руководством преподавателя решать стандартные задачи навыки: - работа под руководством преподавателя на практических занятиях, допустимый уровень культуры исполнения заданий; - достаточный минимальный уровень сформированности заявленных в рабочей программе компетенций; - испытывает затруднения при обосновании алгоритма выполнения заданий</p>
<p>Оценка «неудовлетворительно» (не зачтено)</p>	<p>знания: - фрагментарные знания по дисциплине; - отказ от ответа (выполнения письменной работы); - знание отдельных источников, рекомендованных рабочей программой по дисциплине; умения: - не умеет использовать научную терминологию; - наличие грубых ошибок навыки: - низкий уровень культуры исполнения заданий; - низкий уровень сформированности заявленных в рабочей программе компетенций; - отсутствие навыков самостоятельной работы; - не может обосновать алгоритм выполнения заданий</p>

7.4. Теоретические вопросы и практические задания для проведения промежуточной аттестации обучающихся, необходимые для оценки знаний, умений и навыков и (или) опыта профессиональной деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

7.4.1. Теоретические вопросы для проведения промежуточной аттестации обучающихся  
Теоретические вопросы для промежуточной аттестации (зачет) (семестр 3).

1. Основные типы данных языка C++. Преобразование типов.
2. Основные операции языка C++. Арифметические и логические операции.
3. Основные операции языка C++. Операции ввода/вывода. Форматирование вывода с помощью манипуляторов и флагов.
4. Организаций вычислений в C++. Циклы. Ветвления.
5. Функции. Передача аргументов в функцию по значению.
6. Функции. Передача аргументов в функцию по ссылке.
7. Функции. Передача аргументов в функцию по указателю.
8. Структуры и перечисления.
9. Работа с массивами в C++. Строки как массивы символов.
10. Указатели. Адресная арифметика.
11. Указатели на функцию. Особенности применения.
12. Использование указателей для создания динамических массивов.
13. Многофайловые программы: причины использования, организация и создание.
14. Многофайловые программы. Межфайловое взаимодействие: заголовочные файлы,

библиотечные файлы, пространства имен.

15. Основные функции Win32API для взаимодействия с операционной системой.

Теоретические вопросы для промежуточной аттестации (экзамен) (семестр 4).

1. Парадигмы программирования: предпосылки появления, обзор существующих парадигм программирования.

2. Эволюция процесса создания программного обеспечения. Каскадная модель. Современные подходы.

3. Основные принципы процедурно-ориентированного подхода.

4. Основные принципы объектно-ориентированной парадигмы.

5. Многофайловые программы: причины использования, организация и создание.

6. Многофайловые программы. Межфайловое взаимодействие: межфайловые переменные, межфайловые функции, межфайловые классы.

7. Многофайловые программы. Межфайловое взаимодействие: заголовочные файлы, библиотечные файлы, пространства имен.

8. Объектно-ориентированный анализ: базовые принципы, методология.

9. Средства формализации объектно-ориентированного описания системы. Универсальный язык моделирования (язык UML): назначение, структура, сущности.

10. Язык UML. Виды диаграмм. Диаграммы классов: назначение, используемые элементы, примеры построения.

11. Язык UML. Варианты использования, сценарии. Диаграммы вариантов использования.

12. Понятие класса и объекта в объектно-ориентированном программировании. Члены-данные и члены-функции.

13. Определение класса. Реализация принципа инкапсуляции через управление доступом к компонентам класса.

14. Определение объектов класса. Определение методов класса. Методы класса внутри определения класса. Вызов методов класса. Объекты программы и объекты реального мира. Класс как тип данных.

15. Конструкторы и деструкторы, их роль в классе. Типы конструкторов. Перегрузка конструкторов. Примеры реализаций.

16. Определение методов класса вне класса. Объекты в качестве аргументов функций. Объекты, возвращаемые функцией.

17. Перегрузка методов класса. Структуры и классы: сходство и различие.

18. Классы, объекты, данные и память. Статические и автоматические объекты класса: назначение, особенности и примеры использования.

19. Классы, объекты, данные и память. Константные методы и константные аргументы методов класса. Константные объекты. Массивы объектов.

20. Указатели и объекты. Доступ к членам класса через указатели. Массив указателей на объекты. Сортировка указателей.

21. Использование классов для создания структур для хранения данных. Стек. Преимущества и недостатки, программная реализация.

22. Использование классов для создания структур для хранения данных. Очередь. Преимущества и недостатки, программная реализация.

23. Использование классов для создания структур для хранения данных. Связный список. Преимущества и недостатки, программная реализация.

24. Перегрузка унарных операций: назначение, синтаксические особенности, примеры реализации.

25. Перегрузка бинарных операций: назначение, синтаксические особенности, примеры реализации.

26. Наследование как реализации принципа иерархии в ООП. Виды наследования. Уровни наследования.

27. Реализация принципа наследования в объектно-ориентированных языках программирования. Базовый и производный классы. Конструкторы производного класса.

28. Объявление наследования. Модификация области видимости компонент класса при

наследовании. Перегрузка методов базового класса.

29. Множественное наследование: объявление, примеры реализации.

30. Виртуальные функции и полиморфизм – назначение, примеры практического использования.

31. Абстрактные классы и чистые виртуальные функции: определение, назначение, примеры использования.

32. Дружественные функции, и дружественные классы – определение, назначение, противоречия с основными концепциями ООП. Расширение возможностей по перегрузке операций с помощью дружественных функций.

33. Потокосые классы. Использование потоков для работы с файлами. Форматированный ввод/вывод текстовых файлов.

34. Потокосые классы. Использование потоков для работы с файлами. Ввод/вывод двоичных файлов.

35. Исключительные ситуации. Генерация исключений. Обработка исключений. Блоки повторных попыток (try-блоки), блоки ловушек (catch-блоки).

36. Шаблоны – назначение, примеры использования. Шаблоны функций. Шаблоны классов.

Теоретические вопросы для промежуточной аттестации (экзамен) (семестр 5).

1. Парадигмы программирования: предпосылки появления, обзор существующих парадигм программирования.

2. Эволюция процесса создания программного обеспечения. Каскадная модель. Современные подходы.

3. Основные принципы процедурно-ориентированного подхода.

4. Основные принципы объектно-ориентированной парадигмы.

5. Многофайловые программы: причины использования, организация и создание.

6. Многофайловые программы. Межфайловое взаимодействие: межфайловые переменные, межфайловые функции, межфайловые классы.

7. Многофайловые программы. Межфайловое взаимодействие: заголовочные файлы, библиотечные файлы, пространства имен.

8. Объектно-ориентированный анализ: базовые принципы, методология.

9. Средства формализации объектно-ориентированного описания системы. Универсальный язык моделирования (язык UML): назначение, структура, сущности.

10. Язык UML. Виды диаграмм. Диаграммы классов: назначение, используемые элементы, примеры построения.

11. Язык UML. Варианты использования, сценарии. Диаграммы вариантов использования.

12. Понятие класса и объекта в объектно-ориентированном программировании. Члены- данные и члены-функции.

13. Определение класса. Реализация принципа инкапсуляции через управление доступом к компонентам класса.

14. Определение объектов класса. Определение методов класса. Методы класса внутри определения класса. Вызов методов класса. Объекты программы и объекты реального мира. Класс как тип данных.

15. Конструкторы и деструкторы, их роль в классе. Типы конструкторов. Перегрузка конструкторов. Примеры реализаций.

16. Определение методов класса вне класса. Объекты в качестве аргументов функций. Объекты, возвращаемые функцией.

17. Перегрузка методов класса. Структуры и классы: сходство и различие.

18. Классы, объекты, данные и память. Статические и автоматические объекты класса: назначение, особенности и примеры использования.

19. Классы, объекты, данные и память. Константные методы и константные аргументы методов класса. Константные объекты. Массивы объектов.

20. Указатели и объекты. Доступ к членам класса через указатели. Массив указателей на объекты. Сортировка указателей.

21. Использование классов для создания структур для хранения данных. Стек. Преимущества и недостатки, программная реализация.
22. Использование классов для создания структур для хранения данных. Очередь. Преимущества и недостатки, программная реализация.
23. Использование классов для создания структур для хранения данных. Связный список. Преимущества и недостатки, программная реализация.
24. Перегрузка унарных операций: назначение, синтаксические особенности, примеры реализации.
25. Перегрузка бинарных операций: назначение, синтаксические особенности, примеры реализации.
26. Наследование как реализации принципа иерархии в ООП. Виды наследования. Уровни наследования.
27. Реализация принципа наследования в объектно-ориентированных языках программирования. Базовый и производный классы. Конструкторы производного класса.
28. Объявление наследования. Модификация области видимости компонент класса при наследовании. Перегрузка методов базового класса.
29. Множественное наследование: объявление, примеры реализации.
30. Виртуальные функции и полиморфизм – назначение, примеры практического использования.
31. Абстрактные классы и чистые виртуальные функции: определение, назначение, примеры использования.
32. Дружественные функции, и дружественные классы – определение, назначение, противоречия с основными концепциями ООП. Расширение возможностей по перегрузке операций с помощью дружественных функций.
33. Поточные классы. Использование потоков для работы с файлами. Форматированный ввод/вывод текстовых файлов.
34. Поточные классы. Использование потоков для работы с файлами. Ввод/вывод двоичных файлов.
35. Исключительные ситуации. Генерация исключений. Обработка исключений. Блоки повторных попыток (try-блоки), блоки ловушек (catch-блоки).
36. Шаблоны – назначение, примеры использования. Шаблоны функций. Шаблоны классов.
37. Стандартная библиотека шаблонов (STL). Алгоритмы STL.
38. Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров.
39. Итераторы STL. Адаптеры итераторов. Поточные итераторы. Ассоциативные контейнеры.
40. Основы создания Windows-приложений. Визуальное проектирование. Событийное программирование.
41. Графическая библиотека Microsoft Foundation Classes (MFC). Основные элементы интерфейса.
42. Основные этапы создания Windows-приложения с использованием MFC. Простейшие элементы управления: кнопка, текстовое поле, поле для ввода.
43. Основные этапы создания Windows-приложения с использованием MFC. Функциональные элементы управления: поле со списком, флажок, переключатель.

#### 7.4.2. Практические задания для проведения промежуточной аттестации обучающихся

Практические задания для проведения промежуточной аттестации обучающихся размещены по адресу ЭИОС Moodle (<https://moodle.spbgasu.ru/course/view.php?id=456>)

#### 7.4.3. Примерные темы курсовой работы (проекта) (при наличии)

Типовые контрольные задания и иные материалы текущего контроля успеваемости размещены по адресу ЭИОС Moodle (<https://moodle.spbgasu.ru/course/view.php?id=456>)

#### 7.5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и

(или) опыта профессиональной деятельности, характеризующие этапы формирования компетенций

Процедура проведения промежуточной аттестации и текущего контроля успеваемости регламентируется локальным нормативным актом, определяющим порядок организации и проведения текущего контроля успеваемости и промежуточной аттестации обучающихся.

Процедура оценивания формирования компетенций при проведении текущего контроля приведена в п. 7.2.

Типовые контрольные задания или иные материалы текущего контроля приведены в п. 7.3.

Промежуточная аттестация по дисциплине проводится в форме зачета с оценкой, экзамена.

#### 7.6. Критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии оценивания	Уровень освоения и оценка			
	Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»
	«не зачтено»	«зачтено»		
	Уровень освоения компетенции «недостаточный». Компетенции не сформированы. Знания отсутствуют, умения и навыки не сформированы	Уровень освоения компетенции «пороговый». Компетенции сформированы. Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.	Уровень освоения компетенции «продвинутой». Компетенции сформированы. Знания обширные, системные. Умения носят репродуктивный характер, применяются к решению типовых заданий. Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.	Уровень освоения компетенции «высокий». Компетенции сформированы. Знания аргументированные, всесторонние. Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий. Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка

знания	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> <li>-существенные пробелы в знаниях учебного материала;</li> <li>-допускаются принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий;</li> <li>-непонимание сущности дополнительных вопросов в рамках заданий билета.</li> </ul>	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> <li>-знания теоретического материала;</li> <li>-неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов;</li> <li>-неуверенные и неточные ответы на дополнительные вопросы.</li> </ul>	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> <li>-знание и понимание основных вопросов контролируемого объема программного материала;</li> <li>- знания теоретического материала</li> <li>-способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития;</li> <li>-правильные и конкретные, без грубых ошибок, ответы на поставленные вопросы.</li> </ul>	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> <li>-глубокие, всесторонние и аргументированные знания программного материала;</li> <li>-полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий, в рамках обсуждаемых заданий;</li> <li>-способность устанавливать и объяснять связь практики и теории,</li> <li>-логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора.</li> </ul>
умения	<p>При выполнении практического задания билета обучающийся продемонстрировал недостаточный уровень умений. Практические задания не выполнены. Обучающийся не отвечает на вопросы билета при дополнительных наводящих вопросах преподавателя.</p>	<p>Обучающийся выполнил практическое задание билета с существенными неточностями. Допускаются ошибки в содержании ответа и решении практических заданий. При ответах на дополнительные вопросы было допущено много неточностей.</p>	<p>Обучающийся выполнил практическое задание билета с небольшими неточностями. Показал хорошие умения в рамках освоенного учебного материала. Предложенные практические задания решены с небольшими неточностями. Ответил на большинство дополнительных вопросов.</p>	<p>Обучающийся правильно выполнил практическое задание билета. Показал отличные умения в рамках освоенного учебного материала. Решает предложенные практические задания без ошибок. Ответил на все дополнительные вопросы.</p>

владение навыками	<p>Не может выбрать методику выполнения заданий. Допускает грубые ошибки при выполнении заданий, нарушающие логику решения задач. Делает некорректные выводы. Не может обосновать алгоритм выполнения заданий.</p>	<p>Испытывает затруднения по выбору методики выполнения заданий. Допускает ошибки при выполнении заданий, нарушения логики решения задач. Испытывает затруднения с формулированием корректных выводов. Испытывает затруднения при обосновании алгоритма выполнения заданий.</p>	<p>Без затруднений выбирает стандартную методику выполнения заданий. Допускает ошибки при выполнении заданий, не нарушающие логику решения задач. Делает корректные выводы по результатам решения задачи. Обосновывает ход решения задач без затруднений.</p>	<p>Применяет теоретические знания для выбора методики выполнения заданий. Не допускает ошибок при выполнении заданий. Самостоятельно анализирует результаты выполнения заданий. Грамотно обосновывает ход решения задач.</p>
-------------------	--	---	---	--

Оценка по дисциплине зависит от уровня сформированности компетенций, закрепленных за дисциплиной, и представляет собой среднее арифметическое от выставленных оценок по отдельным результатам обучения (знания, умения, владение навыками).

Оценка «отлично»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 4,5 до 5,0.

Оценка «хорошо»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 3,5 до 4,4.

Оценка «удовлетворительно»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 2,5 до 3,4.

Оценка «неудовлетворительно»/«не зачтено» выставляется, если среднее арифметическое находится в интервале от 0 до 2,4.

## 8. Учебно-методическое и материально-техническое обеспечение дисциплины (модуля)

### 8.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

№ п/п	Автор, название, место издания, издательство, год издания учебной и учебно-методической литературы	Количество экземпляров/электронный адрес ЭБС
<b><u>Основная литература</u></b>		
1	Барков И. А., Объектно-ориентированное программирование, Санкт-Петербург: Лань, 2019	ЭБС
2	Николаев Е. И., Объектно-ориентированное программирование. Часть 2, 2015	<a href="http://www.iprbookshop.ru/63218.html">http://www.iprbookshop.ru/63218.html</a>
3	Иванова Г. С., Ничушкина Т. Н., Объектно-ориентированное программирование, Москва: МГТУ им. Н.Э. Баумана, 2014	ЭБС
4	Николаев Е. И., Объектно-ориентированное программирование. Часть 1, 2015	<a href="http://www.iprbookshop.ru/62966.html">http://www.iprbookshop.ru/62966.html</a>
<b><u>Дополнительная литература</u></b>		
1	Букунов С. В., Букунова О. В., Основы объектно-ориентированного программирования, СПб., 2017	ЭБС
2	Букунов С. В., Основы программирования на языке C++, СПб., 2015	ЭБС



3	Букунов С. В., Букунова О. В., Основы объектно-ориентированного программирования, Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017	<a href="http://www.iprbookshop.ru/74339.html">http://www.iprbookshop.ru/74339.html</a>
4	Лисицин Д. В., Объектно-ориентированное программирование, Новосибирск: Новосибирский государственный технический университет, 2010	<a href="http://www.iprbookshop.ru/44970.html">http://www.iprbookshop.ru/44970.html</a>
5	Сорокин А. А., Объектно-ориентированное программирование, Ставрополь: Северо-Кавказский федеральный университет, 2014	<a href="http://www.iprbookshop.ru/63110.html">http://www.iprbookshop.ru/63110.html</a>
1	Новиков П. В., Объектно-ориентированное программирование, Саратов: Вузовское образование, 2017	<a href="http://www.iprbookshop.ru/64650.html">http://www.iprbookshop.ru/64650.html</a>

Обучающиеся из числа инвалидов и лиц с ОВЗ обеспечиваются печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

8.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
Викиучебник. Объектно-ориентированное программирование [Электронный ресурс]	<a href="http://ru.wikibooks.org/wiki/">http://ru.wikibooks.org/wiki/</a>
Сайт компании Microsoft	<a href="https://msdn.microsoft.com/ru-ru/">https://msdn.microsoft.com/ru-ru/</a>

8.3. Перечень современных профессиональных баз данных и информационных справочных систем

Наименование	Электронный адрес ресурса
Система дистанционного обучения СПбГАСУ Moodle	<a href="https://moodle.spbgasu.ru/">https://moodle.spbgasu.ru/</a>
Электронно-библиотечная система издательства "Лань"	<a href="https://e.lanbook.com/">https://e.lanbook.com/</a>
Электронно-библиотечная система издательства "ЮРАЙТ"	<a href="https://www.biblio-online.ru/">https://www.biblio-online.ru/</a>
Электронно-библиотечная система издательства "IPRbooks"	<a href="http://www.iprbookshop.ru/">http://www.iprbookshop.ru/</a>

8.4. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного и свободно распространяемого программного обеспечения

Наименование	Способ распространения (лицензионное или свободно распространяемое)
Microsoft Office 2016	Договор № Д32009689201 от 18.12.2020г Программные продукты Майкрософт, договор № Д32009689201 от 18.12.2020 с АО "СофтЛайн Трейд": Windows 10, Project Professional 2016, Visio Professional 2016, Office 2016.
Microsoft Visual Studio 2017	Договор № Д32009689201 от 18.12.2020г Программные продукты Майкрософт, договор № Д32009689201 от 18.12.2020 с АО "СофтЛайн Трейд": Windows 10, Project Professional 2016, Visio Professional 2016, Office 2016.

## 8.5. Материально-техническое обеспечение дисциплины

### Сведения об оснащённости учебных аудиторий и помещений для самостоятельной работы

Наименование учебных аудиторий и помещений для самостоятельной работы	Оснащённость оборудованием и техническими средствами обучения
47. Помещения для самостоятельной работы	Помещение для самостоятельной работы (читальный зал библиотеки, ауд. 217): ПК-23 шт., в т.ч. 1 шт.- ПК для лиц с ОВЗ (системный блок, монитор, клавиатура, мышь) с подключением к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду СПбГАСУ. ПО Microsoft Windows 10, Microsoft Office 2016
47. Учебные аудитории для проведения лекционных занятий	Учебная аудитория для проведения занятий лекционного типа, комплект мультимедийного оборудования (персональный компьютер, мультимедийный проектор, экран, аудио-система), доска маркерная белая эмалевая, экран, комплект учебной мебели, подключение к компьютерной сети СПбГАСУ, выход в Интернет.
47. Компьютерный класс	Рабочие места с ПК (стол компьютерный, системный блок, монитор, клавиатура, мышь), стол рабочий, подключение к компьютерной сети СПбГАСУ, выход в Internet.
47. Учебные аудитории для проведения практических занятий, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Учебная аудитория для проведения практических занятий, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации – комплект мультимедийного оборудования (персональный компьютер, мультимедийный проектор, экран, аудио-система), доска маркерная белая эмалевая, комплект учебной мебели, подключение к компьютерной сети СПбГАСУ, выход в Интернет.

Для инвалидов и лиц с ОВЗ обеспечиваются специальные условия для получения образования в соответствии с требованиями нормативно-правовых документов.