



Федеральное государственное бюджетное образовательное учреждение
высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ

Кафедра Информационных технологий

УТВЕРЖДАЮ
Начальник учебно-методического управления

«29» июня 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Разработка программного обеспечения

направление подготовки/специальность 09.03.02 Информационные системы и технологии

направленность (профиль)/специализация образовательной программы Информационные системы и технологии

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 C++ для решения прикладных задач в различных сферах деятельности на базе:

- системного изложения основ алгоритмического языка C++, 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 C++.

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 C++ 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
ОПК-4 Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью с использованием стандартов, норм и правил;	ОПК-4.1 Представляет проект компонента технической документации профессиональной области	знает - основные современные методологии проектирования программного обеспечения - основные CASE-инструменты для проектирования программного обеспечения - основные паттерны проектирования умеет - осуществлять анализ предметной области - разбивать задачу предметной области на абстракции - разрабатывать архитектуру создаваемого программного обеспечения - разрабатывать варианты взаимодействия пользователей с создаваемым программным обеспечением владеет навыками - навыками применения объектно-ориентированного подхода к проектированию программного обеспечения - навыками описания разрабатываемой программной системы с помощью статических и поведенческих диаграмм UML

<p>ОПК-4 Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью с использованием стандартов, норм и правил;</p>	<p>ОПК-4.2 Осуществляет выбор нормативной документации в соответствии с техническим заданием на разработку технической документации</p>	<p>знает</p> <ul style="list-style-type: none"> - основные этапы разработки программного обеспечения. - основные регламенты разработки программного обеспечения <p>умеет</p> <ul style="list-style-type: none"> - читать и понимать техническое задание - осуществлять выбор нормативной документации в соответствии с техническим заданием на разработку технической документации - разрабатывать техническую документацию с использованием современных подходов <p>владеет навыками</p> <ul style="list-style-type: none"> - навыками работы с нормативной документацией - навыками поиска необходимой информации в современных поисковых системах
<p>ОПК-7 Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем;</p>	<p>ОПК-7.1 Осуществляет выбор языка программирования и среды разработки для реализации информационных систем</p>	<p>знает</p> <ul style="list-style-type: none"> - основные парадигмы программирования - основные отличительные особенности популярных языков программирования - основные среды разработки программных и информационных систем <p>умеет</p> <ul style="list-style-type: none"> - осуществлять выбор языка программирования и среды разработки для реализации информационных систем на основе проведения сравнительного анализа <p>владеет навыками</p> <ul style="list-style-type: none"> - навыками программирования на базовых языках высокого уровня - навыками работы в современных средах разработки программных и информационных систем
<p>ОПК-7 Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем;</p>	<p>ОПК-7.2 Составляет требования к аппаратным средствам для реализации информационных систем</p>	<p>знает</p> <ul style="list-style-type: none"> - основные аппаратные средства для разработки программных и информационных систем <p>умеет</p> <ul style="list-style-type: none"> - составлять требования к аппаратным средствам для разработки программных и информационных систем <p>владеет навыками</p> <ul style="list-style-type: none"> - навыками составления требований к аппаратным средствам для разработки программных и информационных систем

<p>ПК-1 Способен проектировать и разрабатывать программное обеспечение</p>	<p>ПК-1.1 Осуществляет выбор метода проектирования программного обеспечения, структуры данных, базы данных или программных интерфейсов из типовых решений и шаблонов</p>	<p>знает</p> <ul style="list-style-type: none"> - основные современные методологии проектирования программного обеспечения - основные CASE-инструменты для проектирования программного обеспечения - основные паттерны проектирования - основные структуры данных - основные системы управления базами данных <p>умеет</p> <ul style="list-style-type: none"> - выбирать метод проектирования программного обеспечения, структуры данных, базы данных или программных интерфейсов из типовых решений и шаблонов <p>владеет навыками</p> <ul style="list-style-type: none"> - навыками применения объектно-ориентированного подхода к проектированию программного обеспечения - навыками описания разрабатываемой программной системы с помощью статических и поведенческих диаграмм UML
<p>ПК-1 Способен проектировать и разрабатывать программное обеспечение</p>	<p>ПК-1.3 Разрабатывает программное обеспечение, структуры данных, базы данных или программные интерфейсы</p>	<p>знает</p> <ul style="list-style-type: none"> - основы программирования - основные структуры данных - основные виды диаграмм UML - основные типы запросов к базе данных - основные графические библиотеки для разработки программных интерфейсов <p>умеет</p> <ul style="list-style-type: none"> - разрабатывать статические и поведенческие диаграммы UML - разрабатывать структуру таблиц базы данных - разрабатывать схемы «сущность-связь» при проектировании базы данных - разрабатывать структуру программного интерфейса <p>владеет навыками</p> <ul style="list-style-type: none"> - навыками разработки программного обеспечения, структур данных, баз данных, программных интерфейсов с использованием современных сред разработки, графических библиотек и систем управления базами данных

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

Данная дисциплина (модуль) включена в Блок «Дисциплины, модули» Б1.О.21 основной профессиональной образовательной программы 09.03.02 Информационные системы и технологии и относится к обязательной части учебного плана.

№ п/п	Предшествующие дисциплины	Код и наименование индикатора достижения компетенции
1	Программирование для ЭВМ	ОПК-3.1, ОПК-6.1, ОПК-6.2
2	Информационные технологии	УК-1.1, УК-1.2, УК-1.6

Программирование для ЭВМ

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.

Информационные технологии

№ п/п	Последующие дисциплины	Код и наименование индикатора достижения компетенции
1	Программирование для Интернет	ПК-1.2, ПК-1.4
2	Программирование на языке Python	ПК-1.3, ПК-1.4

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
Контактная работа	176		48	64	64
Лекционные занятия (Лек)	16	0	16		
Лабораторные занятия (Лаб)	160	160	32	64	64
Иная контактная работа, в том числе:	1,55		0,4	0,65	0,5
консультации по курсовой работе (проекту), контрольным работам (РГР)	1,8		0,4	0,4	1
контактная работа на аттестацию (сдача зачета, зачета с оценкой; защита курсовой работы (проекта); сдача контрольных работ (РГР))	1,05		0,4	0,4	0,25
контактная работа на аттестацию в сессию (консультация перед экзаменом и сдача экзамена)	0,5			0,25	0,25
Часы на контроль	39,5		4	8,75	26,75
Самостоятельная работа (СР)	141,15		55,2	34,2	51,75
Общая трудоемкость дисциплины (модуля)					
часы:	360		108	108	144

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)

№	Разделы дисциплины	Семестр	Контактная работа (по учебным занятиям), час.						СР	Всего, час.	Код индикатора достижения компетенции	
			лекции		ПЗ		ЛР					
			всего	из них на практическую подготовку	всего	из них на практическую подготовку	всего	из них на практическую подготовку				
1.	1 раздел. Основы объектно-ориентированного программирования											
1.1.	Объекты и классы. Конструкторы и деструкторы.	3	2					4	4	7	13	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
1.2.	Перегрузка методов класса. Перегрузка конструкторов. Константные методы класса. Константные объекты.	3	2					4	4	7	13	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
1.3.	Указатели на объекты. Структуры для хранения данных.	3	2					4	4	7	13	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
1.4.	Перегрузка операций.	3	2					4	4	7	13	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
1.5.	Наследование	3	2					4	4	7	13	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
1.6.	Наследование. Включение. Композиция.	3	2					4	4	7	13	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3

1.7.	Работа с файлами.	3	2				4	4	7	13	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
1.8.	Полиморфизм.	3	2				4	4	6,2	12,2	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
2.	2 раздел. Иная контактная работа										
2.1.	Иная контактная работа	3								0,8	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2
3.	3 раздел. Контроль										
3.1.	Зачет	3								4	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.	4 раздел. Создание приложений с использованием стандартных библиотек										
4.1.	Шаблоны и исключения.	4					4	4	2	6	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.2.	Проектирование объектно-ориентированного программного обеспечения. Формализация моделирования. Язык UML.	4					4	4	4,2	8,2	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.3.	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL.	4					4	4	2	6	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.4.	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров.	4					6	6	2	8	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3

4.5.	Итераторы STL. Адаптеры итераторов. Поточковые итераторы.	4					6	6	2	8	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.6.	Контейнеры STL. Ассоциативные контейнеры. Методы контейнеров.	4					6	6	2	8	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.7.	Библиотека WinAPI. Основные функции. Контекст устройства. Создание графических изображений в консольном приложении.	4					4	4	2	6	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.8.	Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Основные этапы создания приложения. Основные функции приложения. Основные элементы управления.	4					6	6	2	8	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.9.	Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Обработка сообщений. Использование растровых изображений. Создание анимации.	4					6	6	4	10	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.10.	Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 1	4					6	6	4	10	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
4.11.	Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 2	4					6	6	4	10	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3

4.12	Создание приложений с использованием библиотeki MFC. Однодокументные приложения. Создание элементов управления. Создание меню. Создание графических изображений.	4					6	6	4	10	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
5.	5 раздел. Иная контактная работа										
5.1.	Иная контактная работа	4								0,8	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
6.	6 раздел. Контроль										
6.1.	Зачет с оценкой	4								9	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.	7 раздел. Основы программирования на языке Java										
7.1.	Введение в Java. Знакомство с JVM, JRE, JDK. Сравнительный анализ Java и других языков программирования.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.2.	Структура программы на языке Java. Импорт пакетов. Консольный ввод/вывод данных.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.3.	Организация вычислений. Циклы. Условные операторы. Использование ссылочных и примитивных типов данных.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.4.	Типы данных в Java.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3

7.5.	Объектно-ориентированные возможности языка Java. Структура класса. Уровни доступа к полям класса. Принципы работы с объектами. Паттерн проектирования Singleton.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.6.	Одномерные и многомерные массивы. Метод clone. Класс Arrays.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.7.	Наследование. Использование конструктора родительского класса внутри дочернего класса.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.8.	Полиморфизм. Абстракции и интерфейсы. Абстрактные классы. Паттерн проектирования Абстрактная фабрика. Паттерн проектирования Фабричный метод.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.9.	Исключения. Использование исключений при разработке собственных библиотек. Stacktrace.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.10	Generic классы. Способы организации. Методы Generic классов.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.11.	Коллекции List. Основные методы. Организация хранения данных с помощью динамического массива и связанного списка. Классы ArrayList и LinkedList.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.12	Коллекции Queue и Stack. Основные методы. Организация хранения данных с помощью динамического массива и связанного списка на базе очереди и стека. Отличие интерфейсов Queue и Stack.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3

7.13	Коллекция HashSet. Коллекция Set. Основные принципы работы с Hash-кодом объекта. Взаимодействие объектов внутри коллекции. Переопределение метода hashCode.	5					4	4	3	7	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.14	Коллекция HashMap. Коллекция Map и ее реализация на основе класса HashMap. Внутреннее устройство коллекции.	5					4	4	4	8	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.15	Коллекции TreeSet. Методы взаимодействия с коллекцией Set. Структуры для хранения данных: бинарное дерево, красно-черное дерево. Внутреннее взаимодействие между объектами внутри коллекции.	5					4	4	4	8	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
7.16	Коллекция TreeMap. Методы взаимодействия с коллекцией Map. Реализация коллекции на основе класса TreeMap. Внутреннее устройство коллекции в формате ключ-значение.	5					4	4	4,75	8,75	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
8.	8 раздел. Иная контактная работа										
8.1.	Иная контактная работа	5								1,25	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3
9.	9 раздел. Контроль										
9.1.	Экзамен	5								27	ОПК-4.1, ОПК-4.2, ОПК-7.1, ОПК-7.2, ПК-1.1, ПК-1.3

5.1. Лекции

№ п/п	Наименование раздела и темы лекций	Наименование и краткое содержание лекций
1	Объекты и классы. Конструкторы и деструкторы.	Объекты и классы. Конструкторы и деструкторы. 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.

2	Перегрузка методов класса. Перегрузка конструкторов. Константные методы класса. Константные объекты.	Перегрузка методов класса. Перегрузка конструкторов. Константные методы класса. Константные объекты. Конструкторы и деструкторы. Структуры и классы. Статические данные класса. Константные методы. Константные объекты. Конструкторы и деструкторы. Перегруженные конструкторы. Определение методов класса вне класса. Аргументы и объекты. Структуры и классы. Статические данные класса. Константные методы. Константные аргументы методов. Константные объекты.
3	Указатели на объекты. Структуры для хранения данных.	Указатели на объекты. Структуры для хранения данных. Указатели на объекты. Массивы указателей. Использование классов для создания структур для хранения данных. Стек. Очередь. Связный список.
4	Перегрузка операций.	Перегрузка операций. Перегрузка операций. Перегрузка унарных операций. Перегрузка бинарных операций. Преобразование типов. Преобразование объектов.
5	Наследование	Наследование Наследование. Базовый и производный классы. Конструкторы производного класса. Базовые функции класса. Иерархия классов. Роль наследования при разработке объектно-ориентированных программ.
6	Наследование. Включение. Композиция.	Наследование. Включение. Композиция. Общее и частное наследование. Уровни наследования. Множественное наследование. Включение как альтернатива наследованию. Композиция как сложное включение.
7	Работа с файлами.	Работа с файлами. Потоки и файлы. Потокосые классы. Потокосый ввод/вывод текстовых файлов. Потокосый ввод/вывод двоичных файлов.
8	Полиморфизм.	Полиморфизм Полиморфизм. Виртуальные функции. Абстрактные классы. Дружественные функции. Дружественные классы. Статические функции.

5.2. Лабораторные работы

№ п/п	Наименование раздела и темы лабораторных работ	Наименование и содержание лабораторных работ
1	Объекты и классы. Конструкторы и деструкторы.	Объекты и классы. Конструкторы и деструкторы. Составление объектно-ориентированных программ с использованием простых классов. Создание конструкторов. Создание деструкторов.
2	Перегрузка методов класса. Перегрузка конструкторов. Константные методы класса. Константные объекты.	Перегрузка методов класса. Перегрузка конструкторов. Константные методы класса. Константные объекты. Составление объектно-ориентированных программ с использованием собственных графических классов.
3	Указатели на объекты. Структуры для хранения данных.	Указатели на объекты. Структуры для хранения данных. Составление объектно-ориентированных программ с использованием массивов указателей. Программирование собственных структур данных: стека, очереди, связного списка.
4	Перегрузка операций.	Перегрузка операций. Составление объектно-ориентированных программ с использованием перегрузки основных арифметических операций и преобразования

		нестандартных типов данных.
5	Наследование	Наследование Составление объектно-ориентированных программ с использованием механизма наследования: создание базовых и производных классов.
6	Наследование. Включение. Композиция.	Наследование. Включение. Композиция. Составление объектно-ориентированных программ с использованием множественного наследования. Практическая реализация таких отношений между классами, как включение и композиция.
7	Работа с файлами.	Работа с файлами. Составление объектно-ориентированных программ, считывающих данные разного типа из текстовых файлов. Составление объектно-ориентированных программ, записывающих данные разного типа в текстовые файлы. Составление объектно-ориентированных программ, считывающих данные разного типа из двоичных файлов. Составление объектно-ориентированных программ, записывающих данные разного типа в двоичные файлы.
8	Полиморфизм.	Полиморфизм Составление объектно-ориентированных программ, использующих абстрактные классы, виртуальные и дружественные функции. Практическая реализация перегрузки методов базового класса в производных классах.
11	Шаблоны и исключения.	Шаблоны и исключения. Составление объектно-ориентированных программ, использующих шаблонные функции и шаблонные классы. Использование классов исключений для обработки исключительных ситуаций.
12	Проектирование объектно-ориентированного программного обеспечения. Формализация моделирования. Язык UML.	Проектирование объектно-ориентированного программного обеспечения. Формализация моделирования. Язык UML. Ознакомление с основными современными подходами к проектированию объектно-ориентированного программного обеспечения. Изучение основ графического языка моделирования UML. Основные типы диаграмм языка UML.
13	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL.	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL. Составление объектно-ориентированных программ, применяющих алгоритмы STL к обычным массивам.
14	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров.	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров. Составление объектно-ориентированных программ, использующих последовательные контейнеры STL. Практическое использование последовательных контейнеров разного типа.
15	Итераторы STL. Адаптеры итераторов. Поточные итераторы.	Итераторы STL. Адаптеры итераторов. Поточные итераторы. Составление объектно-ориентированных программ, использующих итераторы STL разного типа для доступа к данным контейнеров. Практическое применение различных методов контейнеров для обработки данных в контейнерах.
16	Контейнеры STL. Ассоциативные контейнеры. Методы контейнеров.	Контейнеры STL. Ассоциативные контейнеры. Методы контейнеров. Составление объектно-ориентированных программ, использующих ассоциативные контейнеры STL. Практическое использование ассоциативных контейнеров разного типа.
17	Библиотека WinAPI.	Библиотека WinAPI. Основные функции. Контекст устройства.

	<p>Основные функции. Контекст устройства. Создание графических изображений в консольном приложении.</p>	<p>Создание графических изображений в консольном приложении. Составление объектно-ориентированных программ, использующих основные функции библиотеки WinAPI. Создание графических изображений в консольном приложении.</p>
18	<p>Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Основные этапы создания приложения. Основные функции приложения. Основные элементы управления.</p>	<p>Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Основные этапы создания приложения. Основные функции приложения. Основные элементы управления. Создание классических Windows-приложений, использующих основные функции библиотеки WinAPI. Создание графических изображений в оконном приложении. Создание основных элементов управления.</p>
19	<p>Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Обработка сообщений. Использование растровых изображений. Создание анимации.</p>	<p>Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Обработка сообщений. Использование растровых изображений. Создание анимации. Создание классических Windows-приложений, использующих основные функции библиотеки WinAPI. Обработка основных сообщений операционной системы. Использование растровых изображений в оконном приложении. Практическая реализация анимации в окне приложения.</p>
20	<p>Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 1</p>	<p>Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 1 Создание Windows-приложений на основе диалоговых окон с помощью графической библиотеки классов MFC. Практическая реализация диалоговых окон разного типа. Создание простейших элементов управления. Связывание элементов управления с событиями. Обработка событий.</p>
21	<p>Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 2</p>	<p>Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 2 Создание Windows-приложений на основе диалоговых окон с помощью графической библиотеки классов MFC. Увеличение функциональности приложения с помощью элементов управления. Создание пользовательского меню.</p>
22	<p>Создание Windows-приложений с использованием библиотеки MFC. Однодокументные приложения. Создание элементов управления. Создание меню. Создание</p>	<p>Создание Windows-приложений с использованием библиотеки MFC. Однодокументные приложения. Создание элементов управления. Создание меню. Создание графических изображений. Создание однооконных Windows-приложений с помощью графической библиотеки классов MFC. Связывание обработчиков с событиями. Вывод текстовой и числовой информации в окне приложения. Вывод графических изображений в окне приложения. Работа со шрифтами. Создание меню.</p>

	графических изображений.	
25	Введение в Java. Знакомство с JVM, JRE, JDK. Сравнительный анализ Java и других языков программирования.	Введение в Java. Знакомство с JVM, JRE, JDK. Сравнительный анализ Java и других языков программирования. Знакомство с JVM, JRE, JDK. Создание простейших программ на языке Java.
26	Структура программы на языке Java. Импорт пакетов. Консольный ввод/вывод данных.	Структура программы на языке Java. Импорт пакетов. Консольный ввод/вывод данных. Составление программ с использованием пакетов Java. Составление программ, использующих консольный ввод/вывод данных.
27	Организация вычислений. Циклы. Условные операторы. Использование ссылочных и примитивных типов данных.	Организация вычислений. Циклы. Условные операторы. Использование ссылочных и примитивных типов данных. Знакомство с условными операторами в языке Java. Изучение грамотных способов сравнения ссылочных и примитивных типов данных. Понимание правильного выбора цикла для конкретной задачи на языке Java. Составление программ.
28	Типы данных в Java.	Типы данных в Java. Изучение целочисленных, вещественных, символьных и логических примитивных типов данных. Повторение разницы между примитивными и ссылочными типами данных. Понимание принципов работы с объектами. Знакомство со String и Enum. Составление программ.
29	Объектно-ориентированные возможности языка Java. Структура класса. Уровни доступа к полям класса. Принципы работы с объектами. Паттерн проектирования Singleton.	Объектно-ориентированные возможности языка Java. Структура класса. Уровни доступа к полям класса. Принципы работы с объектами. Паттерн проектирования Singleton. Знакомство со структурой класса на языке Java. Понимание разницы между публичными, приватными и защищенными секциями. Изучение правильного выбора полей класса и методов, достаточных для реализации решения задачи. Изучение паттерна проектирования Singleton. Составление программ.
30	Одномерные и многомерные массивы. Метод clone. Класс Arrays.	Одномерные и многомерные массивы. Метод clone. Класс Arrays. Многомерные и одномерные массивы. Способы взаимодействия с одномерными и многомерными массивами. Изучение принципа работы метода clone. Знакомство с классом Arrays и способами его использования при работе с массивами.
31	Наследование. Использование конструктора родительского класса внутри дочернего класса.	Наследование. Использование конструктора родительского класса внутри дочернего класса. Наследование. Принципы грамотного наследования и способов его организации. Изучение методов использования конструктора родительского класса внутри дочернего при наследовании. Составление объектно-ориентированных программ с использованием механизма наследования.
32	Полиморфизм. Абстракции и интерфейсы. Абстрактные классы.	Полиморфизм. Абстракции и интерфейсы. Абстрактные классы. Паттерн проектирования Абстрактная фабрика. Паттерн проектирования Фабричный метод. Составление объектно-ориентированных программ с использованием

	<p>Паттерн проектирования Абстрактная фабрика.</p> <p>Паттерн проектирования Фабричный метод.</p>	<p>полиморфизма, абстрактных классов, паттернов проектирования Абстрактная фабрика и Фабричный метод.</p>
33	<p>Исключения.</p> <p>Использование исключений при разработке собственных библиотек. Stacktrace.</p>	<p>Исключения. Использование исключений при разработке собственных библиотек. Stacktrace.</p> <p>Исключения, Stacktrace. Изучение древа наследования исключений в Java. Понимание уместности и способов “выбрасывания” исключений при неправильной работе программы. Знакомство с принципами использования исключений при разработке собственных библиотек. Понимание устройства Stacktrace. Составление объектно-ориентированных программ с использованием исключений.</p>
34	<p>Generic классы.</p> <p>Способы организации.</p> <p>Методы Generic классов.</p>	<p>Generic классы. Способы организации. Методы Generic классов.</p> <p>Знакомство с Generic классами и способами их организации. Изучение структуры методов у Generic классов. Составление программ с использованием Generic классов.</p>
35	<p>Коллекции List.</p> <p>Основные методы.</p> <p>Организация хранения данных с помощью динамического массива и связанного списка.</p> <p>Классы ArrayList и LinkedList.</p>	<p>Коллекции List. Основные методы. Организация хранения данных с помощью динамического массива и связанного списка. Классы ArrayList и LinkedList.</p> <p>Коллекции List. Знакомство с методами коллекции List. Изучение организации хранения данных с помощью динамического массива и связанного списка. Понимание разницы между классами ArrayList и LinkedList. Составление объектно-ориентированных программ с использованием коллекции List.</p>
36	<p>Коллекции Queue и Stack. Основные методы.</p> <p>Организация хранения данных с помощью динамического массива и связанного списка на базе очереди и стека.</p> <p>Отличие интерфейсов Queue и Stack.</p>	<p>Коллекции Queue и Stack. Основные методы.</p> <p>Организация хранения данных с помощью динамического массива и связанного списка на базе очереди и стека. Отличие интерфейсов Queue и Stack.</p> <p>Коллекции Queue и Stack. Знакомство с методами коллекций Queue и Stack. Изучение организации хранения данных с помощью динамического массива и связанного списка на базе очереди и стека. Понимание разницы между интерфейсами Queue и Stack. Составление объектно-ориентированных программ с использованием коллекций Queue и Stack.</p>
37	<p>Коллекция HashSet.</p> <p>Коллекция Set.</p> <p>Основные принципы работы с Hash-кодом объекта.</p> <p>Взаимодействие объектов внутри коллекции.</p> <p>Переопределение метода hashCode.</p>	<p>Коллекция HashSet. Коллекция Set. Основные принципы работы с Hash-кодом объекта. Взаимодействие объектов внутри коллекции. Переопределение метода hashCode.</p> <p>Коллекция HashSet. Знакомство с коллекцией Set. Изучение принципов работы с Hash-кодом объекта. Понимание внутреннего взаимодействия между объектами внутри коллекции. Знакомство с переопределением метода hashCode. Составление объектно-ориентированных программ с использованием коллекции HashSet.</p>
38	<p>Коллекция HashMap.</p> <p>Коллекция Map и ее реализация на основе класса HashMap.</p> <p>Внутреннее устройство коллекции.</p>	<p>Коллекция HashMap. Коллекция Map и ее реализация на основе класса HashMap. Внутреннее устройство коллекции.</p> <p>Знакомство с коллекцией Map и ее реализацией на основе класса HashMap. Понимание внутреннего устройство коллекции в формате ключ-значение. Повторение важности переопределения метода hashCode при использовании собственных объектов. Составление программ с использованием коллекции HashMap.</p>

39	<p>Коллекции TreeSet. Методы взаимодействия с коллекцией Set. Структуры для хранения данных: бинарное дерево, красно-черное дерево. Внутреннее взаимодействие между объектами внутри коллекции.</p>	<p>Коллекции TreeSet. Методы взаимодействия с коллекцией Set. Структуры для хранения данных: бинарное дерево, красно-черное дерево. Внутреннее взаимодействие между объектами внутри коллекции. Коллекции TreeSet. Повторение методов взаимодействия с коллекцией Set. Изучение таких структур как бинарное и красно- черное дерево. Понимание внутреннего взаимодействия между объектами внутри коллекции. Составление объектно- ориентированных программ с использованием коллекции TreeSet.</p>
40	<p>Коллекция TreeMap. Методы взаимодействия с коллекцией Map. Реализация коллекции на основе класса TreeMap. Внутреннее устройство коллекции в формате ключ-значение.</p>	<p>Коллекция TreeMap. Методы взаимодействия с коллекцией Map. Реализация коллекции на основе класса TreeMap. Внутреннее устройство коллекции в формате ключ-значение. Коллекция TreeMap. Повторение методов взаимодействия с коллекцией Map. Изучение реализации коллекции на основе класса TreeMap. Повторение внутреннего устройства коллекции в формате ключ-значение. Составление объектно-ориентированных программ с использованием коллекции TreeMap.</p>

5.3. Самостоятельная работа обучающихся

№ п/п	Наименование раздела дисциплины и темы	Содержание самостоятельной работы
1	Объекты и классы. Конструкторы и деструкторы.	Объекты и классы. Конструкторы и деструкторы. Изучение материала. Подготовка к практическим занятиям и тестированию.
2	Перегрузка методов класса. Перегрузка конструкторов. Константные методы класса. Константные объекты.	Перегрузка методов класса. Перегрузка конструкторов. Константные методы класса. Константные объекты. Изучение материала. Подготовка к практическим занятиям и тестированию.
3	Указатели на объекты. Структуры для хранения данных.	Указатели на объекты. Структуры для хранения данных. Изучение материала. Подготовка к практическим занятиям и тестированию.
4	Перегрузка операций.	Перегрузка операций. Изучение материала. Подготовка к практическим занятиям и тестированию.
5	Наследование	Наследование Изучение материала. Подготовка к практическим занятиям и тестированию.
6	Наследование. Включение. Композиция.	Наследование. Включение. Композиция. Изучение материала. Подготовка к практическим занятиям и тестированию.
7	Работа с файлами.	Работа с файлами. Изучение материала. Подготовка к практическим занятиям и тестированию.
8	Полиморфизм.	Полиморфизм Изучение материала. Подготовка к практическим занятиям и тестированию.

11	Шаблоны и исключения.	Шаблоны и исключения. Изучение материала. Подготовка к практическим занятиям и тестированию.
12	Проектирование объектно-ориентированного программного обеспечения. Формализация моделирования. Язык UML.	Проектирование объектно-ориентированного программного обеспечения. Формализация моделирования. Язык UML. Изучение материала. Подготовка к практическим занятиям и тестированию.
13	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL.	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL. Изучение материала. Подготовка к практическим занятиям и тестированию.
14	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров.	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров. Изучение материала. Подготовка к практическим занятиям и тестированию.
15	Итераторы STL. Адаптеры итераторов. Поточные итераторы.	Итераторы STL. Адаптеры итераторов. Поточные итераторы. Изучение материала. Подготовка к практическим занятиям и тестированию.
16	Контейнеры STL. Ассоциативные контейнеры. Методы контейнеров.	Контейнеры STL. Ассоциативные контейнеры. Методы контейнеров. Изучение материала. Подготовка к практическим занятиям и тестированию.
17	Библиотека WinAPI. Основные функции. Контекст устройства. Создание графических изображений в консольном приложении.	Библиотека WinAPI. Основные функции. Контекст устройства. Создание графических изображений в консольном приложении. Изучение материала. Подготовка к практическим занятиям и тестированию.
18	Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Основные этапы создания приложения. Основные функции приложения. Основные элементы управления.	Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Основные этапы создания приложения. Основные функции приложения. Основные элементы управления. Изучение материала. Подготовка к практическим занятиям и тестированию.
19	Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Обработка сообщений. Использование растровых изображений. Создание анимации.	Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Обработка сообщений. Использование растровых изображений. Создание анимации. Изучение материала. Подготовка к практическим занятиям и тестированию.

	растровых изображений. Создание анимации.	
20	Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 1	Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 1 Изучение материала. Подготовка к практическим занятиям и тестированию.
21	Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 2	Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 2 Изучение материала. Подготовка к практическим занятиям и тестированию.
22	Создание Windows-приложений с использованием библиотеки MFC. Однодокументные приложения. Создание элементов управления. Создание меню. Создание графических изображений.	Создание Windows-приложений с использованием библиотеки MFC. Однодокументные приложения. Создание элементов управления. Создание меню. Создание графических изображений. Изучение материала. Подготовка к практическим занятиям и тестированию.
25	Введение в Java. Знакомство с JVM, JRE, JDK. Сравнительный анализ Java и других языков программирования.	Введение в Java. Знакомство с JVM, JRE, JDK. Сравнительный анализ Java и других языков программирования. Изучение материала. Подготовка к практическим занятиям и тестированию.
26	Структура программы на языке Java. Импорт пакетов. Консольный ввод/вывод данных.	Структура программы на языке Java. Импорт пакетов. Консольный ввод/вывод данных. Изучение материала. Подготовка к практическим занятиям и тестированию.
27	Организация вычислений. Циклы. Условные операторы. Использование ссылочных и примитивных типов данных.	Организация вычислений. Циклы. Условные операторы. Использование ссылочных и примитивных типов данных. Изучение материала. Подготовка к практическим занятиям и тестированию.
28	Типы данных в Java.	Типы данных в Java. Изучение материала. Подготовка к практическим занятиям и тестированию.
29	Объектно-	Объектно-ориентированные возможности языка Java. Структура

	<p>ориентированные возможности языка Java. Структура класса. Уровни доступа к полям класса. Принципы работы с объектами. Паттерн проектирования Singleton.</p>	<p>класса. Уровни доступа к полям класса. Принципы работы с объектами. Паттерн проектирования Singleton. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
30	<p>Одномерные и многомерные массивы. Метод clone. Класс Arrays.</p>	<p>Одномерные и многомерные массивы. Метод clone. Класс Arrays. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
31	<p>Наследование. Использование конструктора родительского класса внутри дочернего класса.</p>	<p>Наследование. Использование конструктора родительского класса внутри дочернего класса. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
32	<p>Полиморфизм. Абстракции и интерфейсы. Абстрактные классы. Паттерн проектирования Абстрактная фабрика. Паттерн проектирования Фабричный метод.</p>	<p>Полиморфизм. Абстракции и интерфейсы. Абстрактные классы. Паттерн проектирования Абстрактная фабрика. Паттерн проектирования Фабричный метод. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
33	<p>Исключения. Использование исключений при разработке собственных библиотек. Stacktrace.</p>	<p>Исключения. Использование исключений при разработке собственных библиотек. Stacktrace. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
34	<p>Generic классы. Способы организации. Методы Generic классов.</p>	<p>Generic классы. Способы организации. Методы Generic классов. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
35	<p>Коллекции List. Основные методы. Организация хранения данных с помощью динамического массива и связанного списка. Классы ArrayList и LinkedList.</p>	<p>Коллекции List. Основные методы. Организация хранения данных с помощью динамического массива и связанного списка. Классы ArrayList и LinkedList. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
36	<p>Коллекции Queue и Stack. Основные методы. Организация хранения данных с помощью динамического массива и связанного</p>	<p>Коллекции Queue и Stack. Основные методы. Организация хранения данных с помощью динамического массива и связанного списка на базе очереди и стека. Отличие интерфейсов Queue и Stack. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>

	списка на базе очереди и стека. Отличие интерфейсов Queue и Stack.	
37	<p>Коллекция HashSet. Коллекция Set. Основные принципы работы с Hash-кодом объекта. Взаимодействие объектов внутри коллекции. Переопределение метода hashCode.</p>	<p>Коллекция HashSet. Коллекция Set. Основные принципы работы с Hash-кодом объекта. Взаимодействие объектов внутри коллекции. Переопределение метода hashCode. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
38	<p>Коллекция HashMap. Коллекция Map и ее реализация на основе класса HashMap. Внутреннее устройство коллекции.</p>	<p>Коллекция HashMap. Коллекция Map и ее реализация на основе класса HashMap. Внутреннее устройство коллекции. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
39	<p>Коллекции TreeSet. Методы взаимодействия с коллекцией Set. Структуры для хранения данных: бинарное дерево, красно-черное дерево. Внутреннее взаимодействие между объектами внутри коллекции.</p>	<p>Коллекции TreeSet. Методы взаимодействия с коллекцией Set. Структуры для хранения данных: бинарное дерево, красно-черное дерево. Внутреннее взаимодействие между объектами внутри коллекции. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>
40	<p>Коллекция TreeMap. Методы взаимодействия с коллекцией Map. Реализация коллекции на основе класса TreeMap. Внутреннее устройство коллекции в формате ключ-значение.</p>	<p>Коллекция TreeMap. Методы взаимодействия с коллекцией Map. Реализация коллекции на основе класса TreeMap. Внутреннее устройство коллекции в формате ключ-значение. Изучение материала. Подготовка к практическим занятиям и тестированию.</p>

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 зачёту 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.

Работы, выполняемые на практических и лабораторных занятиях, сдаются только лично на занятиях преподавателю, который ведет группу. Задания, выполняемые на компьютере, студенты сначала показывают только в электронном виде в среде разработки MS Visual Studio. При необходимости, в присутствии преподавателя доделывают или исправляют ошибки. Если требуется распечатать выполненные работы и сдать их в бумажном виде, преподаватель говорит об этом на занятиях.

Итогом изучения дисциплины являются зачет, зачет с оценкой, курсовая работа и экзамен. 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	Объекты и классы. Конструкторы и деструкторы.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты

2	Перегрузка методов класса. Перегрузка конструкторов. Константные методы класса. Константные объекты.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
3	Указатели на объекты. Структуры для хранения данных.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
4	Перегрузка операций.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
5	Наследование	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
6	Наследование. Включение. Композиция.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
7	Работа с файлами.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
8	Полиморфизм.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
9	Иная контактная работа	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2	
10	Зачет	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Устный опрос, практическое задание
11	Шаблоны и исключения.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
12	Проектирование объектно-ориентированного программного обеспечения. Формализация моделирования. Язык UML.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
13	Основы работы со стандартной библиотекой шаблонов (STL). Алгоритмы STL.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
14	Контейнеры STL. Виды контейнеров. Последовательные контейнеры. Методы контейнеров.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
15	Итераторы STL. Адаптеры итераторов. Поточковые итераторы.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
16	Контейнеры STL. Ассоциативные контейнеры. Методы контейнеров.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
17	Библиотека WinAPI. Основные функции. Контекст устройства. Создание графических изображений в консольном приложении.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
18	Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Основные этапы создания приложения. Основные функции приложения. Основные	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты

	элементы управления.		
19	Создание оконных приложений с помощью библиотеки WinAPI и среды разработки MS Visual Studio. Обработка сообщений. Использование растровых изображений. Создание анимации.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
20	Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 1	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
21	Создание Windows-приложений в среде Microsoft Visual Studio с помощью графической библиотеки MFC. Приложения на основе диалоговых окон. Часть 2	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
22	Создание Windows- приложений с использованием библиотеки MFC. Однодокументные приложения. Создание элементов управления. Создание меню. Создание графических изображений.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
23	Иная контактная работа	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	
24	Зачет с оценкой	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	
25	Введение в Java. Знакомство с JVM, JRE, JDK. Сравнительный анализ Java и других языков программирования.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
26	Структура программы на языке Java. Импорт пакетов. Консольный ввод/вывод данных.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
27	Организация вычислений. Циклы. Условные операторы. Использование ссылочных и примитивных типов данных.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
28	Типы данных в Java.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
29	Объектно-ориентированные возможности языка Java. Структура класса. Уровни доступа к полям класса. Принципы работы с объектами. Паттерн проектирования Singleton.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
30	Одномерные и многомерные массивы. Метод clone. Класс Arrays.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
31	Наследование. Использование конструктора родительского класса внутри дочернего класса.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
32	Полиморфизм. Абстракции и интерфейсы. Абстрактные классы. Паттерн проектирования Абстрактная	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты

	фабрика. Паттерн проектирования Фабричный метод.		
33	Исключения. Использование исключений при разработке собственных библиотек. Stacktrace.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
34	Generic классы. Способы организации. Методы Generic классов.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
35	Коллекции List. Основные методы. Организация хранения данных с помощью динамического массива и связного списка. Классы ArrayList и LinkedList.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
36	Коллекции Queue и Stack. Основные методы. Организация хранения данных с помощью динамического массива и связного списка на базе очереди и стека. Отличие интерфейсов Queue и Stack.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
37	Коллекция HashSet. Коллекция Set. Основные принципы работы с Hash- кодом объекта. Взаимодействие объектов внутри коллекции. Переопределение метода hashCode.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
38	Коллекция HashMap. Коллекция Map и ее реализация на основе класса HashMap. Внутреннее устройство коллекции.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
39	Коллекции TreeSet. Методы взаимодействия с коллекцией Set. Структуры для хранения данных: бинарное дерево, красно-черное дерево. Внутреннее взаимодействие между объектами внутри коллекции.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
40	Коллекция TreeMap. Методы взаимодействия с коллекцией Map. Реализация коллекции на основе класса TreeMap. Внутреннее устройство коллекции в формате ключ-значение.	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	Практические задания, тесты
41	Иная контактная работа	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 1.3	
42	Экзамен	ОПК-4.1, ОПК-4.2, ОПК- 7.1, ОПК-7.2, ПК-1.1, ПК- 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

Комплект заданий для проверки сформированности индикаторов достижений компетенций ПК-1.1,1.3, ОПК-4.1,4.2,7.1,7.2 размещены в ЭИОС Moodle по адресу <https://moodle.spbgasu.ru/course/view.php?id=3488>

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

<p>Оценка «отлично» (зачтено)</p>	<p>знания: - систематизированные, глубокие и полные знания по всем разделам дисциплины, а также по основным вопросам, выходящим за пределы учебной программы; - точное использование научной терминологии, систематически грамотное и логически правильное изложение ответа на вопросы; - полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю)</p> <p>умения: - умеет ориентироваться в теориях, концепциях и направлениях дисциплины и давать им критическую оценку, используя научные достижения других дисциплин</p> <p>навыки: - высокий уровень сформированности заявленных в рабочей программе компетенций; - владеет навыками самостоятельно и творчески решать сложные проблемы и нестандартные ситуации; - применяет теоретические знания для выбора методики выполнения заданий; - грамотно обосновывает ход решения задач; - безупречно владеет инструментарием учебной дисциплины, умение его эффективно использовать в постановке научных и практических задач; - творческая самостоятельная работа на практических/семинарских/лабораторных занятиях, активно участвует в групповых обсуждениях, высокий уровень культуры исполнения заданий</p>
<p>Оценка «хорошо» (зачтено)</p>	<p>знания: - достаточно полные и систематизированные знания по дисциплине; - усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю)</p> <p>умения: - умеет ориентироваться в основных теориях, концепциях и направлениях дисциплины и давать им критическую оценку; - использует научную терминологию, лингвистически и логически правильно излагает ответы на вопросы, умеет делать обоснованные выводы; - владеет инструментарием по дисциплине, умение его использовать в постановке и решении научных и профессиональных задач</p> <p>навыки: - самостоятельная работа на практических занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий; - средний уровень сформированности заявленных в рабочей программе компетенций; - без затруднений выбирает стандартную методику выполнения заданий; - обосновывает ход решения задач без затруднений</p>

<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 проведения промежуточной аттестации (3 семестр):

1. Основные парадигмы программирования. Достоинства и недостатки. Причины появления.
2. Понятие класса и объекта в объектно-ориентированном программировании. Члены- данные и члены-функции.
3. Определение класса. Реализация принципа инкапсуляции через управление доступом к компонентам класса.
4. Определение объектов класса. Определение методов класса. Методы класса внутри определения класса. Вызов методов класса. Объекты программы и объекты реального мира. Класс как тип данных.
5. Конструкторы и деструкторы, их роль в классе. Типы конструкторов. Перегрузка конструкторов. Примеры реализаций.
6. Определение методов класса вне класса. Объекты в качестве аргументов функций. Объекты, возвращаемые функцией.
7. Перегрузка методов класса. Структуры и классы: сходство и различие.
8. Классы, объекты, данные и память. Статические и автоматические объекты класса: назначение, особенности и примеры использования.

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

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

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

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

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

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

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

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

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

18. Объявление наследования. Модификация области видимости компонент класса при наследовании. Перегрузка методов базового класса.

19. Множественное наследование: объявление, примеры реализации.

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

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

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

23. Поточковые классы. Использование потоков для работы с файлами. Форматированный ввод/вывод текстовых файлов.

24. Поточковые классы. Использование потоков для работы с файлами. Ввод/вывод двоичных файлов.

Теоретические вопросы для проведения промежуточной аттестации (4 семестр):

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

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

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

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

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

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

7. Стандартная библиотека шаблонов (библиотека STL). Алгоритмы STL.

8. Контейнеры STL. Виды контейнеров.

9. Последовательные контейнеры. Особенности организации. Методы контейнеров.

10. Итераторы STL. Назначение. Виды итераторов. Адаптеры итераторов. Поточковые итераторы.

11. Ассоциативные контейнеры STL. Особенности организации. Основные методы.

12. Основы создания Windows-приложений. Визуальное проектирование. Событийное программирование.

13. Библиотека WinAPI. Основные возможности. Основные функции.

14. Библиотека WinAPI. Использование возможностей библиотеки при создании консольных приложений.

15. Библиотека WinAPI. Основные этапы создания классических Windows-приложений.
16. Библиотека WinAPI. Основные функции классических Windows-приложений.
17. Библиотека WinAPI. Основные графические объекты. Принципы создания графических изображений.
18. Библиотека WinAPI. Основные элементы управления. Принципы создания элементов управления в Windows-приложениях.
19. Библиотека WinAPI. Основные подходы к обработке сообщений. Типы сообщений.
20. Библиотека WinAPI. Использование растровых изображений. Создание анимации.
21. Графическая библиотека MFC. Основные этапы создания MFC-приложения.
22. Графическая библиотека MFC. Основные элементы интерфейса.
23. Графическая библиотека MFC. Основные виды приложений.
24. Графическая библиотека MFC. Приложения на основе диалоговых окон. Этапы создания. Основные особенности.
25. Графическая библиотека MFC. Приложения на основе диалоговых окон. Основные функции.
26. Графическая библиотека MFC. Приложения на основе диалоговых окон. Простейшие элементы управления: кнопка, текстовое поле, поле для ввода.
27. Графическая библиотека MFC. Приложения на основе диалоговых окон. Функциональные элементы управления: поле со списком, флажок, переключатель.
28. Графическая библиотека MFC. Однодокументные (SDI) приложения. Этапы создания. Основные особенности.
29. Графическая библиотека MFC. Однодокументные (SDI) приложения. Основные классы.
30. Графическая библиотека MFC. Однодокументные (SDI) приложения. Создание элементов управления. Обработка событий.
31. Графическая библиотека MFC. Однодокументные (SDI) приложения. Создание графических изображений.

Теоретические вопросы для проведения промежуточной аттестации (5 семестр):

1. Особенности языка Java. JVM, JRE, JDK. Сравнительный анализ Java и других языков программирования.
2. Структура программы на языке Java. Импорт пакетов. Консольный ввод/вывод.
3. Организация вычислений. Циклы. Условные операторы. Использование ссылочных и примитивных типов данных.
4. Типы данных языка Java.
5. Объектно-ориентированные возможности языка Java. Структура класса. Уровни доступа к полям класса. Принципы работы с объектами. Паттерн проектирования Singleton.
6. Одномерные и многомерные массивы. Метод clone. Класс Arrays и способы его использования при работе с массивами.
7. Наследование. Принципы грамотного наследования и способы его организации. Использование конструктора родительского класса внутри дочернего класса.
8. Полиморфизм. Абстракции и интерфейсы. Абстрактные классы. Абстрактные интерфейсы. Паттерн проектирования Абстрактная фабрика. Паттерн проектирования Фабричный метод.
9. Исключения. Дерево наследования исключений в Java. Использование исключений при разработке собственных библиотек. Stacktrace.
10. Generic классы. Способы организации. Методы Generic классов.
11. Коллекции List. Основные методы. Организация хранения данных с помощью динамического массива и связанного списка. Классы ArrayList и LinkedList.
12. Коллекции Queue и Stack. Основные методы. Организация хранения данных с помощью динамического массива и связанного списка на базе очереди и стека. Отличие интерфейсов Queue и Stack.
13. Коллекция HashSet. Коллекция Set. Основные принципы работы с Hash-кодом объекта. Взаимодействие объектов внутри коллекции. Переопределение метода hashCode.
14. Коллекция HashMap. Коллекция Map и ее реализация на основе класса HashMap. Внутреннее устройство коллекции.

15. Коллекции TreeSet. Методы взаимодействия с коллекцией Set. Структуры для хранения данных: бинарное дерево, красно-черное дерево. Внутреннее взаимодействие между объектами внутри коллекции.

16. Коллекция TreeMap. Методы взаимодействия с коллекцией Map. Реализация коллекции на основе класса TreeMap. Внутреннее устройство коллекции в формате ключ-значение.

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=3488>)

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

Курсовой проект

Разработка собственной библиотеки

Описание:

Вам предстоит разработать систему классов (один или несколько) для реализации задач, связанных с темой вашего варианта. При разработке вам разрешено пользоваться такими готовыми классами, как:

- Имплементации интерфейсов Queue, Stack, List, Set и Map;
- Arrays;
- String ;
- StringBuilder и StringBuffer.

Требования:

- Библиотека должна решать большинство задач, связанных с вашим вариантом (Можете проконсультироваться с преподавателем).
- Классы должны иметь как статичные, так и нестатичные поля и методы.
- Система классов, по возможности, должна демонстрировать основные принципы объектно-ориентированного программирования и содержать Generic классы.
- Количество методов не ограничено, но минимум их должно быть 10.

Варианты заданий:

1. Библиотека для работы с матрицами;
2. Библиотека для работы с графами;
3. Библиотека для решения ЗЛП.
4. Библиотека для парсинга сайтов (по полученному html);
5. Библиотека для построения графиков

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, зачета с оценкой и 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»		

	<p>Уровень освоения компетенции «недостаточный». Компетенции не сформированы. Знания отсутствуют, умения и навыки не сформированы</p>	<p>Уровень освоения компетенции «пороговый». Компетенции сформированы. Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.</p>	<p>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.</p>	<p>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</p>
знания	<p>Обучающийся демонстрирует: -существенные пробелы в знаниях учебного материала; -допускаются принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; -непонимание сущности дополнительных вопросов в рамках заданий билета.</p>	<p>Обучающийся демонстрирует: -знания теоретического материала; -неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; -неуверенные и неточные ответы на дополнительные вопросы.</p>	<p>Обучающийся демонстрирует: -знание и понимание основных вопросов контролируемого объема программного материала; - знания теоретического материала -способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития; -правильные и конкретные, без грубых ошибок, ответы на поставленные вопросы.</p>	<p>Обучающийся демонстрирует: -глубокие, всесторонние и аргументированные знания программного материала; -полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий, в рамках обсуждаемых заданий; -способность устанавливать и объяснять связь практики и теории, -логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора.</p>

<p>умения</p>	<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>
<p>владение навыками</p>	<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
<u>Основная литература</u>		
1	Букунов С. В., Букунова О. В., Создание объектно-ориентированных приложений с использованием стандартных библиотек классов. Библиотека STL, СПб., 2018	http://ntb.spbgasu.ru/elib/00914/
2	Лисицин Д. В., Объектно-ориентированное программирование, Новосибирск: Новосибирский государственный технический университет, 2010	http://www.iprbookshop.ru/44970.html
3	Букунов С. В., Букунова О. В., Основы объектно-ориентированного программирования, СПб., 2017	http://ntb.spbgasu.ru/elib/00799/
4	Джошуа Блох, Стрельцов В., Усманов Р., Java. Эффективное программирование, Саратов: Профобразование, 2017	http://www.iprbookshop.ru/64057.html
5	Сорокин А. А., Объектно-ориентированное программирование, Ставрополь: Северо-Кавказский федеральный университет, 2014	http://www.iprbookshop.ru/63110.html
6	Вязовик Н. А., Программирование на Java, Москва: Интернет- Университет Информационных Технологий (ИНТУИТ), 2016	http://www.iprbookshop.ru/73710.html
<u>Дополнительная литература</u>		
1	Тузовский А. Ф., Объектно-ориентированное программирование, Москва: Юрайт, 2022	https://urait.ru/bcode/490369
2	Николаев Е. И., Объектно-ориентированное программирование. Часть 2, , 2015	http://www.iprbookshop.ru/63218.html
3	Николаев Е. И., Объектно-ориентированное программирование. Часть 1, , 2015	http://www.iprbookshop.ru/62966.html
4	Барков И. А., Объектно-ориентированное программирование, Санкт-Петербург: Лань, 2019	https://e.lanbook.com/book/119661
1	Новиков П. В., Объектно-ориентированное программирование, Саратов: Вузовское образование, 2017	http://www.iprbookshop.ru/64650.html

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]	http://ru.wikibooks.org/wiki/
Сайт компании Microsoft	https://msdn.microsoft.com/ru-ru/

8.3. Перечень современных профессиональных баз данных и информационных справочных систем

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

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.

<p>47. Учебные аудитории для проведения практических занятий, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации</p>	<p>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.</p>
---	--

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