



Федеральное государственное бюджетное образовательное учреждение
высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ

Кафедра Информатики

УТВЕРЖДАЮ
Начальник учебно-методического управления

«31» октября 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Основы программирования

направление подготовки/специальность 38.03.05 Бизнес-информатика

направленность (профиль)/специализация образовательной программы Бизнес-аналитика

Форма обучения очная

Санкт-Петербург, 2024

1. Цели и задачи освоения дисциплины (модуля)

Целью освоения дисциплины "Основы программирования" является:

- развитие базовых навыков программирования, позволяющих обучающимся понимать и реализовывать принципы создания алгоритмов и составления программ;
- изучение основных возможностей языков программирования Python и C++ и их применение при разработке программ.

Задачи дисциплины:

- ознакомление с теоретическими основами программирования;
- изучение основ алгоритмизации;
- развитие навыков объектно-ориентированного и функционального программирования;
- формирование навыков создания приложений на языках программирования Python и C++;
- освоение современных сред создания программных продуктов.

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-3 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.1 Осуществляет управление процессами разработки и использования продуктов и услуг в сфере информационно-коммуникационных технологий	знает – современные технологии и инструменты разработки продуктов и услуг в сфере информационно-коммуникационных технологий; умеет – разрабатывать стратегию управления жизненным циклом продукта в сфере информационно-коммуникационных технологий; владеет – навыками работы с инструментами для оценки эффективности процессов разработки и использования информационно-коммуникационных технологий.
ОПК-3 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.2 Применяет алгоритмы, языки программирования и управления базами данных, современные программные среды разработки информационных систем для решения прикладных бизнес-задач	знает – методику составления алгоритмов для решения прикладных бизнес-задач и оценки их эффективности; умеет – выбирать подходящий метод, алгоритм для решения поставленной задачи; владеет – навыками разработки алгоритмов для решения прикладных бизнес-задач и их программных реализаций.

<p>ОПК-3 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации</p>	<p>ОПК-3.3 Разрабатывает алгоритмы и программный код для практического применения в информационных системах управления бизнесом</p>	<p>знает</p> <ul style="list-style-type: none"> – основные алгоритмические структуры и методы программирования; – языки программирования Python и C++; – принципы разработки программного обеспечения; – технологии баз данных и принципы работы с ними; <p>умеет</p> <ul style="list-style-type: none"> – анализировать требования пользователей; – разрабатывать и оптимизировать алгоритмы для решения конкретных задач; – создавать и тестировать программный код; – интегрировать разработанные решения в существующие информационные системы; <p>владеет</p> <ul style="list-style-type: none"> – инструментами для разработки и отладки программного обеспечения.
---	---	--

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

Данная дисциплина (модуль) включена в Блок «Дисциплины, модули» Б1.О.15 основной профессиональной образовательной программы 38.03.05 Бизнес-информатика и относится к обязательной части учебного плана.

Требования к предварительной подготовке обучающегося

знать:

- основные понятия информатики, основы алгоритмизации и программирования в соответствии с образовательной программой среднего общего образования;
- иметь представление о современных средствах вычислительной техники;
- основные разделы курса математики в соответствии с образовательной программой среднего общего образования;

уметь:

- использовать возможности вычислительной техники и программного обеспечения;
- составлять алгоритмы решения простых задач в рамках курса информатики, соответствующего образовательной программе среднего общего образования;

владеть:

- первичными навыками и основными методами решения математических задач;
- навыками работы с персональным компьютером и компьютерными сетями;
- навыками работы с учебной литературой.

№ п/п	Последующие дисциплины	Код и наименование индикатора достижения компетенции
1	Анализ данных на языке Python	ОПК-4.1, ОПК-4.2, ОПК-4.3
2	Введение в машинное обучение	ОПК-3.1, ОПК-3.2, ОПК-3.3
3	Защита информации	ОПК-3.1, ОПК-3.3, ОПК-5.1
4	Управление базами данных	ОПК-3.1, ОПК-3.2, ОПК-3.3, ОПК-5.1
5	Информационные технологии финансового и экономического анализа	ПК-2.3, ПК-4.1, ПК-1.1, ПК-1.2
6	Информационно-аналитическая деятельность поддержки принятия решений	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	
			2	3
Контактная работа	112		64	48
Лекционные занятия (Лек)	32	0	16	16
Лабораторные занятия (Лаб)	80	0	48	32
Иная контактная работа, в том числе:	0,5		0,25	0,25
консультации по курсовой работе (проекту), контрольным работам (РГР)				
контактная работа на аттестацию (сдача зачета, зачета с оценкой; защита курсовой работы (проекта); сдача контрольных работ (РГР))				
контактная работа на аттестацию в сессию (консультация перед экзаменом и сдача	0,5		0,25	0,25
Часы на контроль	17,5		8,75	8,75
Самостоятельная работа (СР)	122		71	51
Общая трудоемкость дисциплины (модуля)				
часы:	252		144	108
зачетные единицы:	7		4	3

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. Введение в программирование. Основы программирования на языке Python										
1.1.	Введение в программирование	2	2				6	8	16	ОПК-3.1, ОПК-3.2, ОПК-3.3	
1.2.	Условные операторы и циклы	2	2				6	9	17	ОПК-3.1, ОПК-3.2, ОПК-3.3	
1.3.	Коллекции	2	2				6	9	17	ОПК-3.1, ОПК-3.2, ОПК-3.3	
1.4.	Функции	2	2				6	9	17	ОПК-3.1, ОПК-3.2, ОПК-3.3	
1.5.	Введение в объектно-ориентированное программирование	2	2				6	9	17	ОПК-3.1, ОПК-3.2, ОПК-3.3	

1.6.	Создание графических приложений	2	2				6		9	17	ОПК-3.1, ОПК-3.2, ОПК-3.3
1.7.	Разработка Web-приложения на Python	2	2				6		9	17	ОПК-3.1, ОПК-3.2, ОПК-3.3
1.8.	Работа с базами данных в Python	2	2				6		9	17	ОПК-3.1, ОПК-3.2, ОПК-3.3
2.	2 раздел. Контроль										
2.1.	Зачет с оценкой	2								9	ОПК-3.1, ОПК-3.2, ОПК-3.3
3.	3 раздел. Основы программирования на C++										
3.1.	Введение в программирование на C++	3	2				4		7	13	ОПК-3.1, ОПК-3.2, ОПК-3.3
3.2.	Синтаксис C++. Средства описания данных и средства описания действий	3	2				4		8	14	ОПК-3.1, ОПК-3.2, ОПК-3.3
3.3.	Структурированные типы данных в C++	3	2				4		8	14	ОПК-3.1, ОПК-3.2, ОПК-3.3
3.4.	Процедурное программирование в C++	3	2				4		8	14	ОПК-3.1, ОПК-3.2, ОПК-3.3
3.5.	Объектно-ориентированное программирование на C++	3	6				12		13	31	ОПК-3.1, ОПК-3.2, ОПК-3.3
3.6.	Объектно-ориентированный анализ и проектирование	3	2				4		7	13	ОПК-3.1, ОПК-3.2, ОПК-3.3
4.	4 раздел. Контроль										
4.1.	Зачет с оценкой	3								9	ОПК-3.1, ОПК-3.2, ОПК-3.3

5.1. Лекции

№ разд	Наименование раздела и темы лекций	Наименование и краткое содержание лекций
1	Введение в программирование	Введение в программирование Компилируемые и интерпретируемые языки программирования. Среды разработки, онлайн компиляторы. Ввод / вывод информации. Типы данных и арифметические действия. Переменная и ее место в памяти.
2	Условные операторы и циклы	Условные операторы и циклы Условные операторы. Блок if-elif-else. Операции сравнения и логические операции. Вложенные условия. Цикл с предусловием и постусловием. Операции break, continue. Вложенные циклы. Цикл for. Примеры решения задач.
3	Коллекции	Коллекции

		Операции со списками и со строками. Индексация списков. Хранение данных, ввод данных. Список списков. Хранение данных в памяти компьютера. Срезки строк и списков.
4	Функции	Функции Понятие функции и функционального программирования. Аргументы функции. Область видимости переменных. Возвращаемые значения. Вызов функции в памяти компьютера. Лямбда функции.
5	Введение в объектно-ориентированное программирование	Введение в объектно-ориентированное программирование Понятие объектно-ориентированного программирования (ООП). Класс, экземпляр и метод класса. Основные принципы ООП. Полиморфизм, инкапсуляция, наследование.
6	Создание графических приложений	Создание графических приложений Графический интерфейс пользователя (GUI). Событийно-ориентированное программирование. Инструменты для создания графического интерфейса пользователя: модуль стандартной библиотеки tkinter, фреймворк PyQt.
7	Разработка Web-приложения на Python	Взаимодействие с интернетом. Разработка web-приложений Основы web-программирования. Фреймворки для разработки web-приложений: Flask, Django.
8	Работа с базами данных в Python	Работа с базами данных в Python Основы SQLite. Доступ к базам данных SQLite из Python. Выполнение запросов, обработка результатов. Управление транзакциями.
10	Введение в программирование на C++	Введение в программирование на C++ Язык C++, обзор. Структура программ. Компиляция
11	Синтаксис C++. Средства описания данных и средства описания действий	Синтаксис C++. Средства описания данных и средства описания действий Типы данных, переменные, операции и выражения, преобразования типов. Условные операторы и циклы. Ввод-вывод в C++.
12	Структурированные типы данных в C++	Структурированные типы данных в C++ Указатели и массивы. Организация работы с динамической памятью. Реализация динамических структур данных.
13	Процедурное программирование в C++	Процедурное программирование в C++ Блоки. Функции. Передача параметров в функции. Рекурсивные вызовы. Перегрузка функций. Шаблоны функций.
14	Объектно-ориентированное программирование на C++	Основы и механизмы реализации объектно-ориентированного программирования на C++ Объектно-ориентированное программирование. Инкапсуляция, наследование, полиморфизм. Описание абстрактных типов данных, доступ к компонентам класса. Методы класса. Конструкторы и деструкторы. Перегрузка операций. Реализация наследования и полиморфизма. Шаблоны классов. Стандартная библиотека шаблонов STL.
15	Объектно-ориентированный анализ и проектирование	Объектно-ориентированный анализ и проектирование Анализ требований и объектный анализ. Принципы проектирования. Итерации и тестирование.

5.2. Лабораторные работы

№ разд	Наименование раздела и темы лабораторных работ	Наименование и содержание лабораторных работ
1	Введение в программирование	Типы данных. Базовые функции для работы с различными типами данных Типы данных: числовые типы, строки, списки, кортежи, множества, диапазоны, словари. Базовые функции для работы с различными типами данных.
2	Условные операторы и циклы	Условные операторы и циклы. Обработка исключений Условные операторы и циклы. Обработка исключений. Решение задач.
3	Коллекции	Операции со списками и строками Индексация списков. Хранение данных, ввод данных. Список списков. Хранение данных в памяти компьютера. Срезки строк и списков.
4	Функции	Встроенные и пользовательские функции Встроенные и пользовательские функции. Итераторы и функции-генераторы. Декораторы функций.
4	Функции	Основы программирования модулей Модули. Основы программирования модулей. Модули стандартной библиотеки.
4	Функции	Работа с файлами. Модули для работы с файлами Работа с файлами. Модули для работы с файлами.
5	Введение в объектно-ориентированное программирование	Объектно-ориентированное программирование. Основы программирования классов Создание экземпляров класса. Конструктор и деструктор класса. Перегрузка операторов.
5	Введение в объектно-ориентированное программирование	Объектно-ориентированное программирование. Наследование. Композиция. Абстрактные методы классов. Декораторы классов.
6	Создание графических приложений	Разработка оконного приложения Модуль tkinter. Класс Tk. Основные виджеты, упаковщики. Привязка событий.
7	Разработка Web-приложения на Python	Разработка web-приложения Web-фреймворк Flask. Разработка web-приложения.
8	Работа с базами данных в Python	Доступ к базам данных SQLite из Python Работа с базой данных. Язык запросов SQL. Библиотека Sqlite.
10	Введение в программирование на C++	Введение в программирование на C++ Структура языка программирования C++. Базовые элементы языка C++: алфавит, лексемы, выражения, комментарии. Предложения языка: описания и операторы. Программа на языке C++: состав и структура.
11	Синтаксис C++. Средства описания данных и средства описания действий	Синтаксис C++. Средства описания данных и средства описания действий Арифметические типы: целочисленные, с плавающей точкой. Логический тип bool. Тип void. Идентификаторы языка. Область действия и область видимости идентификатора. Ключевые (зарезервированные) слова. Константы. Разделители. Базовые конструкции структурного

		программирования: средства реализации линейных, разветвляющихся, циклических алгоритмов.
12	Структурированные типы данных в C++	Структурированные типы данных в C++ Структурированные типы данных: одномерные и многомерные массивы. Решение типовых задач на массивы. Указатели и ссылки. Динамические массивы. Динамические структуры: списки, стеки, очереди.
13	Процедурное программирование в C++	Процедурное программирование в C++ Принципы использования функций в программах. Решение задач на примере реализации алгоритмов сортировки.
14	Объектно-ориентированное программирование на C++	Основы и механизмы реализации объектно-ориентированного программирования на C++ Разработка классов. Составляющие классов: поля, методы. Экземпляры класса. Перегрузка унарных и бинарных операций, операции вызова функции, операции индексирования. Наследование. Создание и использование шаблонов классов.
15	Объектно-ориентированный анализ и проектирование	Объектно-ориентированный анализ и проектирование Основы доказательства правильности программ. Теорема структуры и структурное программирование. Анализ программ. Корректность программ. Способы верификации программ. Жизненный цикл программы.

5.3. Самостоятельная работа обучающихся

№ разд	Наименование раздела дисциплины и темы	Содержание самостоятельной работы
1	Введение в программирование	Введение в программирование Изучение материала. Подготовка к лабораторным работам и тестированию.
2	Условные операторы и циклы	Условные операторы и циклы Изучение материала. Подготовка к лабораторным работам и тестированию.
3	Коллекции	Коллекции Изучение материала. Подготовка к лабораторным работам и тестированию.
4	Функции	Функции Изучение материала. Подготовка к лабораторным работам и тестированию.
5	Введение в объектно-ориентированное программирование	Введение в объектно-ориентированное программирование Изучение материала. Подготовка к лабораторным работам и тестированию.
6	Создание графических приложений	Создание графических приложений Изучение материала. Подготовка к лабораторным работам и тестированию.
7	Разработка Web-приложения на Python	Разработка Web-приложения на Python Изучение материала. Подготовка к лабораторным работам и тестированию.
8	Работа с базами	Работа с базами данных в Python

	данных в Python	Изучение материала. Подготовка к лабораторным работам и тестированию.
10	Введение в программирование на C++	Введение в программирование на C++ Изучение материала. Подготовка к лабораторным работам и тестированию.
11	Синтаксис C++. Средства описания данных и средства описания действий	Синтаксис C++. Средства описания данных и средства описания действий Изучение материала. Подготовка к лабораторным работам и тестированию.
12	Структурированные типы данных в C++	Структурированные типы данных в C++ Изучение материала. Подготовка к лабораторным работам и тестированию.
13	Процедурное программирование в C++	Процедурное программирование в C++ Изучение материала. Подготовка к лабораторным работам и тестированию.
14	Объектно-ориентированное программирование на C++	Объектно-ориентированное программирование на C++ Изучение материала. Подготовка к лабораторным работам и тестированию.
15	Объектно-ориентированный анализ и проектирование	Объектно-ориентированный анализ и проектирование Изучение материала. Подготовка к лабораторным работам и тестированию.

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

Программой дисциплины предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал, а также 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;
- подготовка к лабораторным занятиям;
- подготовка к текущему контролю успеваемости студентов;
- подготовка к зачету с оценкой.

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 практических 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 практические 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 лабораторных занятиях, сдаются только лично на занятиях преподавателю, который ведет группу. Задания, выполняемые на компьютере, обучающиеся сначала показывают только в электронном виде в соответствующих программах. При необходимости, при преподавателе доделывают или исправляют ошибки. Если требуется распечатать выполненные работы и сдать их в бумажном виде, преподаватель говорит об этом на занятиях.

Итогом изучения дисциплины является зачет с оценкой. Зачет с оценкой проводится по расписанию. Форма проведения занятия может быть устная, письменная и в электронном виде. Обучающиеся, 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 в программирование	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
2	Условные операторы и циклы	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
3	Коллекции	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты

4	Функции	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
5	Введение в объектно-ориентированное программирование	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
6	Создание графических приложений	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
7	Разработка Web-приложения на Python	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
8	Работа с базами данных в Python	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
9	Зачет с оценкой	ОПК-3.1, ОПК-3.2, ОПК-3.3	
10	Введение в программирование на C++	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
11	Синтаксис C++. Средства описания данных и средства описания действий	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
12	Структурированные типы данных в C++	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
13	Процедурное программирование в C++	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
14	Объектно-ориентированное программирование на C++	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
15	Объектно-ориентированный анализ и проектирование	ОПК-3.1, ОПК-3.2, ОПК-3.3	Практические задания, тесты
16	Зачет с оценкой	ОПК-3.1, ОПК-3.2, ОПК-3.3	

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 ОПК-3.1, ОПК-3.2, ОПК-3.3

Тестовые и контрольные задания размещены: ЭИОС / СДО СПбГАСУ Moodle / Кафедры / Информатики / БАКАЛАВРИАТ и СПЕЦИАЛИТЕТ - кафедра информатики / Основы программирования (<https://moodle.spbgasu.ru/course/view.php?id=3054>)

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">- 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>умения:</p> <ul style="list-style-type: none">- умеет ориентироваться в теориях, концепциях и направлениях дисциплины и давать им критическую оценку, используя научные достижения других дисциплин <p>навыки:</p> <ul style="list-style-type: none">- 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>Оценка «хорошо» (зачтено)</p>	<p>знания:</p> <ul style="list-style-type: none">- достаточно полные и систематизированные знания по дисциплине;- усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю) <p>умения:</p> <ul style="list-style-type: none">- 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>навыки:</p> <ul style="list-style-type: none">- 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>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

Примерные вопросы

Раздел 1. Введение в программирование. Основы программирования на языке Python

1. Язык программирования Python: описание и общие сведения.
2. Исходный код на языке Python.
3. Выражения, идентификаторы, пространства имен и области видимости.
4. Управляющие конструкции: операторы выбора и цикла.
5. Обработка исключений.
6. Функции. Лямбда-выражения.
7. Встроенные типы: целочисленный, вещественный, комплексный, логический.
8. Кортежи. Списки. Срезы. Словари. Множества и операции над ними.
9. Файлы и работа с ними.
10. Модули и пакеты. Стандартные модули (sys, os, math, random).
11. Функции преобразования типов.
12. Функции ввода-вывода.
13. Функциональное программирование. Виды параметров функций. Параметры по умолчанию.
14. Декораторы.

15. Генераторы.
16. Итераторы.
17. Объектно-ориентированное программирование. Объявление класса.
18. Атрибуты, свойства, сокрытие данных.
19. Типизация и полиморфизм.
20. Перегрузка операторов.
21. Наследование.
22. Инструменты для создания графического интерфейса пользователя.
23. Разработка web-приложений на Pytjon.
24. Работа с базами данных в Python.

Раздел 2. Основы программирования на C++

1. Составные части программы на языке C++.
2. Константы и их типы. Типы и размерность переменных.
3. Ввод-вывод данных.
4. Операции. Приоритеты операций..
5. Условные операторы.
6. Операторы цикла.
7. Массив, работа с массивом.
8. Указатель, работа с указателем.
9. Функции, передача параметров в функции.
10. Обработка символьных строк.
11. Сложные типы данных, структуры.
12. Работа с файлами, текстовые и бинарные файлы.
13. Функции стандартной библиотеки. Математические и тригонометрические функции.
14. ООП. Доступ к компонентам класса. Методы класса.
15. Конструкторы и деструкторы.
16. Перегрузка операций.
17. Реализация наследования и полиморфизма.
18. Шаблоны классов. Стандартная библиотека шаблонов STL.

7.4.2. Практические задания для проведения промежуточной аттестации обучающихся

Практические задания размещены: ЭИОС / СДО СПбГАСУ Moodle / Кафедры / Информатики / БАКАЛАВРИАТ и СПЕЦИАЛИТЕТ - кафедра информатики / Основы программирования (<https://moodle.spbgasu.ru/course/view.php?id=3054>)

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

Курсовые работы (проекты) учебным планом не предусмотрены.

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. 7.3. Типовые контрольные задания или иные материалы текущего контроля приведены в п. 7.2. Промежуточная аттестация по дисциплине проводится в форме зачета с оценкой. Зачет с оценкой проводится в форме практического контрольного задания или тестового задания и собеседования.

7.6. Критерии оценивания сформированности компетенций при проведении промежуточной аттестации

	Уровень освоения и оценка			
	Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»
	«не зачтено»	«зачтено»		
Критерии оценивания	<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 «продвинутый». 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>

<p>знания</p>	<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., Основы объектно-ориентированного программирования, Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017	http://www.iprbookshop.ru/74339.html
2	Букунов С. В., Основы программирования на языке C++, Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2015	http://www.iprbookshop.ru/63631.html
3	Букунов С. В., Букунова О. В., Объектно ориентированное программирование на языке Python, Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2020	https://www.iprbookshop.ru/117194.html

4	Букунов С. В., Букунова О. В., Разработка приложений с графическим пользовательским интерфейсом на языке Python, Санкт-Петербург: Лань, 2023	https://e.lanbook.com/book/292856
<u>Дополнительная литература</u>		
1	Демяненко Я. М., Чердынцева М. И., Программирование на языке C++, Москва: Ай Пи Ар Медиа, 2022	https://www.iprbooks.hop.ru/128007.html
2	Титов А. Н., Тазиева Р. Ф., Введение в Tkinter. Разработка графических интерфейсов в Python, Казань: Издательство КНИТУ, 2023	https://www.iprbooks.hop.ru/136144.html
3	Чернышев С. А., Основы программирования на Python, Москва: Юрайт, 2024	https://urait.ru/bcode/544190
4	Огнева М. В., Кудрина Е. В., Программирование на языке C++: практический курс, Москва: Юрайт, 2024	https://urait.ru/bcode/539713
5	Федоров Д. Ю., Программирование на языке высокого уровня Python, Москва: Юрайт, 2024	https://urait.ru/bcode/539651

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
Система дистанционного обучения СПбГАСУ Moodle	https://moodle.spbgasu.ru/

8.3. Перечень современных профессиональных баз данных и информационных справочных систем

Наименование	Электронный адрес ресурса
Образовательные интернет-ресурсы СПбГАСУ	https://www.spbgasu.ru/university/obrazovatelnye-internet-resursy/
Российская государственная библиотека	www.rsl.ru
Научная электронная библиотека eLIBRARY.RU	https://www.elibrary.ru/
Электронно-библиотечная система издательства "IPRsmart"	http://www.iprbookshop.ru/
Электронно-библиотечная система издательства "ЮРАЙТ"	https://www.biblio-online.ru/
Электронно-библиотечная система издательства "Лань"	https://e.lanbook.com/
Электронная библиотека Ирбис 64	http://ntb.spbgasu.ru/irbis64r_plus/

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)
Python версия 3.7.6386.10	Свободно распространяемое
Apache NetBeans	Свободно распространяемое
PyCharm Community	Свободно распространяемое
Visual Studio Community Edition	Свободно распространяемое
Dbeaver	Свободно распространяемое
DB Browser for SQLite	Свободно распространяемое
Anaconda	Свободно распространяемое

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
72. Учебные аудитории для проведения лекционных занятий	Учебная аудитория для проведения занятий лекционного типа, комплект мультимедийного оборудования (персональный компьютер, мультимедийный проектор, экран, аудиосистема), доска маркерная белая эмалевая, экран, комплект учебной мебели, подключение к компьютерной сети СПбГАСУ, выход в Интернет.
72. 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, аудиосистема), доска маркерная белая эмалевая, комплект учебной мебели, подключение к компьютерной сети СПбГАСУ, выход в Интернет.
72. Компьютерный класс	Рабочие места с ПК (стол компьютерный, системный блок, монитор, клавиатура, мышь), стол рабочий, подключение к компьютерной сети СПбГАСУ, выход в Internet.
72. 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.

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

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 38.03.05 Бизнес-информатика (приказ Минобрнауки России от 29.07.2020 № 838).

Программу составил:
доцент Инф, к.ф.-м.н. Мовсесова Л.В.

Программа обсуждена и рекомендована на заседании кафедры Информатики
30.08.2024, протокол № 1
Заведующий кафедрой Мовсесова Лия Витальевна

Программа одобрена на заседании учебно-методической комиссии факультета
19.09.2024, протокол № 2.

Председатель УМК д.э.н., профессор Г.Ф. Токунова

**Тестовые задания для проверки сформированности индикаторов достижения компетенций
ОПК-3.1, ОПК-3.2, ОПК-3.3.****Раздел 1. Основы программирования на языке Python****Вопрос 1.** Какая функция преобразует строку в список?

- list()
- eval()
- tuple()
- repr()

Вопрос 2. Какая функция преобразует символ в число?

- oct()
- hex()
- ord()
- unichr()

Вопрос 3. Какой оператор возвращает истину, если он не находит переменную в указанной последовательности, и ложь в противном случае?

- not in
- is
- //
- **

Вопрос 4. Какая функция определяет принадлежность символов нижнему регистру?

- istitle()
- isnumeric()
- islower()
- isspace()

Вопрос 5. Какая из переменных среды является альтернативным путем поиска модуля?

- PYTHONHOME
- PYTHONCASEOK
- PYTHONPATH

– PYTHONSTARTUP

Вопрос 6. Какая функция преобразует объект в строку?

– str()

– long()

– float()

– int()

Вопрос 7. Какой оператор выполняет целочисленное деление?

– %

– /

– //

– ни один из перечисленных

Вопрос 8. Чувствителен ли Python к регистру при работе с идентификаторами?

Да / Нет

Вопрос 9. Какая функция перегружает оператор + ?

– __sum__() Да

– __plus__()

– __add__()

– ни одна из перечисленных

Вопрос 10. Что из перечисленного является кортежем?

– {}

– [1, 2, 3]

– {1, 2, 3}

– (1, 2, 3)

Вопрос 11. Что получится в результате вычисления выражения $3*1**3$?

– 1

– 3

– 9

– 27

Вопрос 12. Какое ключевое слово используется для создания класса?

- create
- main
- class
- declare

Вопрос 13. Какой элемент отвечает за инициализацию полей класса?

- метод
- объект
- конструктор
- интерфейс

Вопрос 14. Что можно считать описание реализации общей задачи в ООП?

- метод
- класс
- процедура
- функция

Раздел 2. Основы программирования на C++

Вопрос 1. Какая команда используется для передачи управления из функции к вызывающей функции?

- go back
- switch
- return
- goto

Вопрос 2. Какая стандартная функция используется, чтобы найти последние соответствия символа в строке?

- strrchr()
- strnstr()
- strstr()
- aststr()

Вопрос 3. Верно ли следующее утверждение: первый аргумент в командной строке всегда должен показывать общее количество аргументов?

Вопрос 4. Какое утверждение о шаблонах классов неверно?

- До инстанцирования шаблона класса компилятор не может провести полную проверку данного шаблона на наличие семантических ошибок.
- Шаблон класса может иметь в качестве параметров, как некоторые типы, так и константы.
- Если два класса созданы на основе одного и того же шаблона классов, то это не означает, что они связаны между собой отношением наследования.
- Класс, созданный на основе шаблона, не является потомком этого шаблона.
- Конкретные классы на основе шаблонов создаются на этапе выполнения программы.

Вопрос 5. Какой стандартный алгоритм позволяет применить некоторую операцию ко всем элементам последовательности?

- for_each
- forall
- for_all
- while
- foreach

Вопрос 6. Какого типа полиморфизма не существует?

- статического
- абстрактного
- динамического
- параметрического
- ad-hoc

Вопрос 7. Какой тип описывает конструкция `int(*p[8])()`; ?

- Указатель на массив функций, принимающих `void` и возвращающих `int`.
- Эта конструкция не корректна в C++.
- Указатель на функцию, возвращающую массив с элементами типа `int`.
- Массив указателей на функцию, принимающую `void` и возвращающую `int`.
- Объявление функции, возвращающей массив указателей на `int`.

Вопрос 8. Область видимости переменной, объявленной внутри блока, распространяется...

- На весь файл, содержащий код с объявлением переменной.
- От начала блока до его конца.
- От места объявления переменной до конца блока.
- На все пространство имен.
- На всю функцию, содержащую объявление данной переменной.

Вопрос 9. С помощью какого оператора можно вычислить размер операнда в байтах?

- `size_of()`
- `sizeof()`
- `size()`
- `SIZEOF()`
- `_sizeof(operand)`

Вопрос 10. Оператор typeid() возвращает...

- Идентификатор типа.
- Указатель на массив символов (char*), содержащий имя типа.
- Ссылку на тип стандартной библиотеки, называемый type_info.
- Ссылку на метакласс.
- Идентификатор класса.

Раздел 1. Основы программирования на языке Python

Примеры заданий

Задание 1. Дан файл, содержащий текст на русском языке. Выполнить задание в зависимости от номера варианта.

- Определить сколько раз встречаются заданные слова в тексте; сколько раз они встречаются непосредственно друг за другом.
- Найти слово, содержащее наибольшее количество указанных букв.
- Подсчитать, сколько слов начинается с указанной буквы.
- Найти слово, встречающееся в каждом предложении, или сообщить, что такого слова нет.
- Зашифровать текст, заменив каждую букву на следующую за ней по алфавиту.

Задание 2. Создать базу данных по заданной теме (в зависимости от номера варианта). БД должна содержать не менее трех таблиц. Реализовать возможность экспорта/импорта из/в xml-файл, а также предусмотреть не менее трех статистических запросов.

Раздел 2. Основы программирования на C++

Примеры заданий

Задание 1. Разработать программу реализующую работу с комплексными числами:

- сложение двух комплексных чисел.
- вычитание двух комплексных чисел.
- печать комплексного числа в форме (a, b), где a – действительная часть, b – мнимая часть числа.

Задание 2. Разработать программу реализующую работу с файлами:

- разработать функцию чтения двумерного массива из файла в динамическую память и нахождения максимальных элементов каждой строки;
- уменьшить каждый элемент массива на найденное значение максимального элемента и записать полученный новый массив в файл.