

**Филиал федерального государственного бюджетного образовательного учреждения
высшего образования
«Национальный исследовательский университет «МЭИ»
в г. Смоленске**

УТВЕРЖДАЮ

Зам. директора
по учебно-методической работе
филиала ФГБОУ ВО
«НИУ «МЭИ» в г. Смоленске
В.В. Рожков
«20» 08 2020 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ

(НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ)

Направление подготовки: **09.03.01 «Информатика и вычислительная техника»**

Профиль **«Автоматизированные системы обработки информации и управления»**

Уровень высшего образования: **бакалавриат**

Нормативный срок обучения: **4 года**

Форма обучения: **очная**

Год набора: **2020**

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины: изучение языков и методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию, как языков программирования, так и методов программирования

Задачи:

- изучение особенности программирования, основы языка программирования C++ и Python;
- применение программирования на практике при разработке программного обеспечения на языке программирования C++ и Python;
- рассмотрение классификации программных средств и возможности их применения для решения практических задач;
- проведение анализа технической документации по использованию программного средства.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Программирование» относится к обязательной части программы.

Перечень последующих дисциплин, для которых необходимы знания, умения и навыки, формируемые данной дисциплиной: «Математическая логика и теория алгоритмов».

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины направлено на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОП ВО по данному направлению подготовки (*специальности*):

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

Компетенция	Индикаторы достижения компетенций	Результаты обучения
ОПК-7. Способен участвовать в настройке и наладке программно-аппаратных комплексов	ОПК-7.1	Знает: специфику установки программного обеспечения, связанной с типом, разрядностью операционной системы; Умеет: устанавливать программное обеспечение с учетом особенностей операционных систем; Владеет: навыками написания программного обеспечения и последующей установки.
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1	Знает: особенности программирования; основы языка программирования C++ и Python; Умеет: применять программирование на практике при разработке программного обеспечения на языке программирования C++ и Python; Владеет: основами программирования на языке C++ и

		Python, навыками отладки и тестирования прототипов программно-технических комплексов задач.
ОПК-9. Способен осваивать методики использования программных средств для решения практических задач	ОПК-9.1	Знает: классификацию программных средств и возможности их применения для решения практических задач; Умеет: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи; Владеет: способами описания методики использования программного средства для решения конкретной задачи в виде технической документации.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Структура дисциплины:

Индекс	Наименование	Семестр 1										Семестр 2										Итого за курс										
		Контроль	Академических часов									з.е.	Контроль	Академических часов									з.е.	Контроль	Академических часов						з.е.	
			Всего	Контакт.	Лек	Лаб	Пр	КРП	СР	Контроль	Всего			Контакт.	Лек	Лаб	Пр	КРП	СР	Контроль	Всего	Контакт.			Лек	Лаб	Пр	КРП	СР	Контроль		Всего
Б1.О.08	Программирование	Экз	252	68	34	34				148	36	7	Экз	180	50	34	16			94	36	5	Экз(2)	432	118	68	50			242	72	12

ОБОЗНАЧЕНИЯ:

Виды промежуточной аттестации (виды контроля):

Экз – экзамен;

ЗаО – зачет с оценкой;

За – зачет;

Виды работ:

Контакт. – контактная работа обучающихся с преподавателем;

Лек. – лекционные занятия;

Лаб.– лабораторные работы;

Пр. – практические занятия;

КРП – курсовая работа (курсовой проект);

РГР – расчетно-графическая работа (реферат);

СР – самостоятельная работа студентов;

з.е. – объем дисциплины в зачетных единицах.

Содержание дисциплины в осеннем семестре 2020/2021:

№	Наименование видов занятий и тематик, содержание
1	<p>Лекционные занятия 17 шт. по 2 часа:</p> <p>1.1. Введение в язык. Основные понятия и история языка. Структура программы. Объявления (фундаментальные типы, идентификаторы, литералы, неявные конверсии типов, разница между объявлением и определением, области видимости). Выражения (базовые операторы, операторы присваивания, операторы сравнения, тернарный оператор, инкремент и декремент, оператор sizeof, оператор запятая).</p> <p>1.2. Управляющие конструкции. Виды ошибок и UB (compile-time error (ошибки компиляции), runtime error (ошибки выполнения), undefined behaviour (UB)). Модификаторы типов и связанные с ними вопросы. Указатели. Функции (указатели на функции, функции с переменным числом аргументов, аргументы по умолчанию, перегрузка функций).</p> <p>1.3. Операторы new и delete. Ссылки. Константы. Операторы явного преобразования типов (cast). Операторы «.», «->», this. Оператор присваивания, правило трех. Списки инициализации в конструкторах. Ключевое слово explicit. Константные методы.</p> <p>1.4. Ключевое слово mutable. Ключевое слово friend. Делегирующие конструкторы. Ключевое слово static. Указатели на члены.</p> <p>1.5. Перегрузка операторов. Арифметические операторы. Операторы сравнения. Инкремент и декремент. Квадратные скобки. Круглые скобки. Операторы «&&», « » и «,». Операторы «*», «&» и «- > ». Перегрузка C-style cast.</p> <p>1.6. Введение в ООП. Классы и структуры. Конструктор, деструктор и оператор присваивания (конструктор, деструктор, конструктор копирования).</p> <p>1.7. Наследование. Объявление. Соккрытие имен при наследовании. Поиск имен при наследовании (соккрытие имен наследником, явный вызов методов предка, почему компилятор может не увидеть публичный метод или атрибут класса, различие классов и структур при наследовании, тонкости запретов при public-наследовании от класса с private-наследованием, доступность друзьям protected методов/атрибутов предка).</p> <p>1.8. Порядок вызова конструкторов и деструкторов при наследовании. Множественное наследование (проблема ромбовидного наследования, проблема «сын - прямой наследник бабушки»). Виртуальное наследование.</p> <p>1.9. Приведение типов между наследниками (dynamic_cast). Виртуальные функции. Виртуальный деструктор. Абстрактные классы. Override и final. Оператор typeid.</p> <p>1.10. Шаблоны. Объявление шаблонов. Специализации шаблонов. Использование typedef, typename, remove_const, remove_reference, etc. Правила вывода шаблонных типов. non-type template parameters.</p> <p>1.11. Variadic templates (шаблоны с переменным количеством шаблонных аргументов). Использование функтора. CRTP (Curious Recurring Template Pattern).</p> <p>1.12. Исключения (exceptions). Особенности исключений. Ключевые слова throw, try, catch. Отличие между исключениями и ошибками. Последовательность и правила обработки исключений.</p> <p>1.13. Копирование при исключениях. Проблемы исключений (спецификации исключений «старая версия» и «новая версия»). Проблема исключений в конструкторах. Проблема исключений в деструкторах. Гарантия безопасности при исключениях.</p> <p>1.14. Аллокаторы. Оператор placement-new. Перегрузка operator new и operator delete. nothrow new, new_handler, allocator, allocator_traits.</p> <p>1.15. Контейнеры. Последовательные контейнеры (vector, deque, array, list, forward_list)</p>

	<p>Методы insert, splice, unique, sort, erase, size.</p> <p>1.16. Ассоциативные контейнеры (set, multiset, map, multimap). Методы find, count, lower_bound, upper_bound, equal_range, unordered_map, unordered_set, unordered_multimap. Адаптеры (stack, queue, priority_queue).</p> <p>1.17. Шаблонные параметры. hash. Специальные функции (для хеш-таблиц). Сравнение времени работы операций для контейнеров.</p>
2	<p>Лабораторные работы 11 шт. по 2 часа:</p> <p>2.1. Линейные алгоритмы.</p> <p>2.2. Разветвляющиеся алгоритмы.</p> <p>2.3. Циклические алгоритмы.</p> <p>2.4. Циклические алгоритмы, практическое применение.</p> <p>2.5. Циклические алгоритмы, обработка последовательностей.</p> <p>2.6. Двухмерные массивы. Обработка части массива.</p> <p>2.7. Функции.</p> <p>2.8. Функции. Массивы, как параметры.</p> <p>2.9. Рекурсия.</p> <p>2.10. Файлы.</p> <p>2.11. Классы.</p> <p>Лабораторные работы 3 шт. по 4 часа:</p> <p>2.12. Одномерные массивы.</p> <p>2.13. Двухмерные массивы</p> <p>2.14. Строки.</p>
3	<p>Самостоятельная работа студентов:</p> <p>3.1. Защита лабораторных работ 2.1-2.14;</p> <p>3.2. 5 контрольных опросов после 5-й, 9-й, 11-й, 13-й и 17-й лекций;</p> <p>3.3. Подготовка к экзамену по дисциплине (оценочные материалы приведены в разделе 6 настоящей РПД)</p>

Содержание дисциплины в весеннем семестре 2020/2021:

№	Наименование видов занятий и тематик, содержание
1	<p>Лекционные занятия 17 шт. по 2 часа:</p> <p>1.1. Система контроля версий git. Основные операции, ветки, коммиты. Язык программирования python: описание и философия. Нотификация типов.</p> <p>1.2. Стиль написания: yagni, kiss, dry, solid. Кодировка, физические и логические строки. Операторные скобки. Юникод и эмодзи.</p> <p>1.3. Строки, строковые литералы, операции со строками. Операции форматирования. Идентификаторы, пространства имен, область видимости.</p> <p>1.4. Встроенные типы в языке программирования Python: целочисленный, вещественный, комплексный, логический. Управляющие конструкции: операторы ветвлений и циклов.</p> <p>1.5. Функции в языке Python, lambda функции. Функции первого рода, чистые функции. Виды параметров функции. Параметры по умолчанию: *args, **kwargs.</p> <p>1.6. Функции преобразования типов. Функции ввода-вывода.</p> <p>1.7. Последовательности. Кортежи. (,) и tuple(). Последовательности. Списки. Срезы. [] и list(). Последовательности. Словари. {} и dict(). Множества и операции над ними. set().</p> <p>1.8. Обработка исключений. Декораторы. Генераторы. Генераторные выражения.</p> <p>1.9. Итераторы. Функции: iter, enumerate, sorted. Модуль itertools. Функции itertools.chain, itertools.repeat, itertools.count. Функции для обработки последовательностей:</p>

	<p>range, map, filter, sum, re-duce, zip.</p> <p>1.10. Объектно-ориентированное программирование (ООП) на языке программирования Python. Описание классов. ООП в python: атрибуты, свойства, сокрытие данных.</p> <p>1.11. Типизация и полиморфизм. Имитация типов. Перегрузка операторов. Магические методы.</p> <p>1.12. ООП в Python. Особенности наследования. Статические методы и методы класса.</p> <p>1.13. Утиная типизация в Python. Менеджер контекста. Работа с файлами в Python. Чтение бесконечных файлов.</p> <p>1.14. Асинхронное программирование. Мотивация, проблемы и решения. Asyncio.async/await. Корутины. GIL и многопоточные программы. Мьютексы и семафоры. Класс Thread. Поток, процесс.</p> <p>1.15. Модули и пакеты. GUI библиотеки. Tkinter, PyQt, Библиотеки numpy, matplotlib, pandas, sympy.</p> <p>1.16. Фреймворки Django, aiohttp. Особенности применения. Тестирование. Юнит и интеграционное тестирование. Форматы CSV, json, xml. Чтение и запись средствами Python.</p> <p>1.17. Регулярные выражения. Определение, шаблоны, основные методы. Ускорение Python. Numba, Cython, интеграция с C/C++.</p>
2	<p>Лабораторные работы 8 шт. по 2 часа:</p> <p>2.1. Система контроля версий git.</p> <p>2.2. Структура проекта. Обсуждение технического задания.</p> <p>2.3. Разработка библиотеки</p> <p>2.4. Выгрузка библиотек в открытый доступ.</p> <p>2.5. Тестирование.</p> <p>2.6. Код ревью.</p> <p>2.7. Оформление пояснительной записки.</p> <p>2.8. Представление проекта.</p>
3	<p>Самостоятельная работа студентов:</p> <p>3.1. Защита лабораторных работ 2.1-2.8;</p> <p>3.2. 5 контрольных опросов после 5-й, 9-й, 11-й, 13-й и 17-й лекций;</p> <p>3.3. Подготовка к экзамену по дисциплине (оценочные материалы приведены в разделе 6 настоящей РПД)</p>

Текущий контроль: контрольные опросы после 5-й, 9-й, 11-й, 13-й и 17-й лекций, устный опрос на каждой лекции по материалам предыдущей пары, защита лабораторных работ.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Таблица – Образовательные технологии, используемые при реализации различных видов учебной занятости по дисциплине

№ п/п	Виды учебных занятий	Образовательные технологии
1	Лекции	Классическая (традиционная, информационная) лекция
2	Лабораторная работа	Технология выполнения лабораторных заданий индивидуально

		Допуск к лабораторной работе
3	Самостоятельная работа студентов (внеаудиторная)	Информационно-коммуникационные технологии (доступ к ЭИОС филиала, к ЭБС филиала, доступ к информационно-методическим материалам по дисциплине)
4	Контроль (промежуточная аттестация: экзамен)	Технология устного опроса

6. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ – ДЛЯ ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ДИСЦИПЛИНЫ

К промежуточной аттестации студентов по дисциплине могут привлекаться представители работодателей, преподаватели последующих дисциплин, заведующие кафедрами.

Оценка качества освоения дисциплины включает как текущий контроль успеваемости, так и промежуточную аттестацию.

Оценочные средства текущего контроля успеваемости:

Примеры вопросов к контрольному опросу после 5-й лекции:

1. Назовите основные понятия языка программирования.
2. Что включают в себя базовые операторы?
3. Что включают в себя операторы присваивания?
4. Что включают в себя операторы сравнения?
5. Что включают в себя тернарный оператор?
6. Что включают в себя инкремент и декремент?
7. Что включают в себя оператор sizeof и оператор запятая?
8. Опишите модификаторы типов.
9. Раскройте понятие операторов явного преобразования типов (cast), операторов new и delete.

Примеры вопросов к контрольному опросу после 9-й лекции:

1. Опишите ключевые слова mutable, friend, static.
2. В чем заключается перезагрузка операторов?
3. Какие арифметические операторы, операторы сравнения Вам известны?
4. Что означает ООП? Опишите его классы и структуры.
5. Почему компилятор может не увидеть публичный метод или атрибут класса?

Примеры вопросов к контрольному опросу после 11-й лекции:

1. Опишите порядок выполнения вызова конструкторов и деструкторов при наследовании.
2. Раскройте проблему ромбовидного наследования.
3. Раскройте проблемы «сын – прямой наследник бабушки».
4. В чем заключаются виртуальные функции?
5. В чем заключаются операторы override, final и typeid.

Примеры вопросов к контрольному опросу после 13-й лекции:

1. Раскройте понятие шаблона, опишите его объявление и специализации.
2. Для чего используются typedef, typename, remove_const, remove_reference, etc.

3. Опишите правила вывода шаблонных типов.
4. В чем заключаются шаблоны с переменным количеством шаблонных аргументов?

Примеры вопросов к контрольному опросу после 17-й лекции:

1. Раскройте понятие исключения, выявите его особенности.
2. Опишите ключевые слова `throw`, `try`, `catch`.
3. Опишите последовательность и правила обработки исключений.
4. В чем заключается проблема исключений в деструкторах?
5. Как можно гарантировать безопасность при исключениях?
6. В чем заключается оператор `placement-new`.
7. Что такое контейнеры? Раскройте понятие и опишите его классификацию.

Оценочные средства для промежуточной аттестации:

Примеры вопросов к экзамену по дисциплине в осеннем семестре 2020/2021:

1. Введение в язык. Основные понятия и история языка. Структура программы. Объявления (фундаментальные типы, идентификаторы, литералы, неявные конверсии типов, разница между объявлением и определением, области видимости).
2. Выражения (базовые операторы, операторы присваивания, операторы сравнения, тернарный оператор, инкремент и декремент, оператор `sizeof`, оператор запятой).
3. Управляющие конструкции. Виды ошибок и UB (`compile-time error` (ошибки компиляции), `runtime error` (ошибки выполнения), `undefined behaviour (UB)`).
4. Модификаторы типов и связанные с ними вопросы. Указатели. Функции (указатели на функции, функции с переменным числом аргументов, аргументы по умолчанию, перегрузка функций).
5. Операторы `new` и `delete`. Ссылки. Константы. Операторы явного преобразования типов (`cast`). Операторы «`», «->», this`. Оператор присваивания, правило трех. Списки инициализации в конструкторах. Ключевое слово `explicit`. Константные методы.
6. Ключевое слово `mutable`. Ключевое слово `friend`. Делегирующие конструкторы. Ключевое слово `static`. Указатели на члены.
7. Перегрузка операторов. Арифметические операторы. Операторы сравнения. Инкремент и декремент. Квадратные скобки. Круглые скобки. Операторы «`&&`», «`||`» и «`,`». Операторы «`*`», «`&`» и «`- >`». Перегрузка C-style `cast`.
8. Введение в ООП. Классы и структуры. Конструктор, деструктор и оператор присваивания (конструктор, деструктор, конструктор копирования).
9. Наследование. Объявление. Соккрытие имен при наследовании. Поиск имен при наследовании (соккрытие имен наследником, явный вызов методов предка, почему компилятор может не увидеть публичный метод или атрибут класса, различие классов и структур при наследовании, тонкости запретов при `public`-наследовании от класса с `private`-наследованием, доступность друзьям `protected` методов/атрибутов предка).
10. Порядок вызова конструкторов и деструкторов при наследовании. Множественное наследование (проблема ромбовидного наследования, проблема «сын - прямой наследник бабушки»). Виртуальное наследование.
11. Приведение типов между наследниками (`dynamic_cast`). Виртуальные функции. Виртуальный деструктор. Абстрактные классы. `Override` и `final`. Оператор `typeid`.
12. Шаблоны. Объявление шаблонов. Специализации шаблонов. Использование `typedef`, `typename`, `remove_const`, `remove_reference`, etc. Правила вывода шаблонных типов. `non-type template parameters`.
13. `Variadic templates` (шаблоны с переменным количеством шаблонных аргументов). Использование функтора. CRTP (`Curious Recurring Template Pattern`).

14. Исключения (exceptions). Особенности исключений. Ключевые слова throw, try, catch. Отличие между исключениями и ошибками. Последовательность и правила обработки исключений.

15. Копирование при исключениях. Проблемы исключений (спецификации исключений «старая версия» и «новая версия»). Проблема исключений в конструкторах. Проблема исключений в деструкторах. Гарантия безопасности при исключениях.

16. Аллокаторы. Оператор placement-new. Перегрузка operator new и operator delete. nothrow new, new_handler, allocator, allocator_traits.

17. Контейнеры. Последовательные контейнеры (vector, deque, array, list, forward_list) Методы insert, splice, unique, sort, erase, size.

18. Ассоциативные контейнеры (set, multiset, map, multimap). Методы find, count, lower_bound, upper_bound, equal_range, unordered_map, unordered_set, unordered_multimap.

19. Адаптеры (stack, queue, priority_queue).

20. Шаблонные параметры. hash. Специальные функции (для хеш-таблиц). Сравнение времени работы операций для контейнеров.

Примеры вопросов к экзамену по дисциплине в весеннем семестре 2020/2021:

1. Система контроля версий git. Основные операции, ветки, коммиты.
2. Язык программирования Python: описание и философия. Нотификация типов.
3. Стилль написания: yagni, kiss, dry, solid.
4. Кодировка, физические и логические строки. Операторные скобки. Юникод и эмодзи.
5. Строки, строковые литералы, операции со строками. Операции форматирования.
6. Идентификаторы, пространства имен, область видимости.
7. Встроенные типы в языке программирования Python: целочисленный, вещественный, комплексный, логический.
8. Управляющие конструкции: операторы ветвлений и циклов.
9. Функции в языке Python, lambda функции. Функции первого рода, чистые функции.
10. Виды параметров функции. Параметры по умолчанию: *args, **kwargs.
11. Функции преобразования типов.
12. Функции ввода-вывода.
13. Последовательности. Кортежи. (,) и tuple().
14. Последовательности. Списки. Срезы. [] и list().
15. Последовательности. Словари. {} и dict().
16. Множества и операции над ними. set().
17. Обработка исключений.
18. Декораторы.
19. Генераторы. Генераторные выражения.
20. Итераторы. Функции: iter, enumerate, sorted.
21. Модуль itertools. Функции itertools.chain, itertools.repeat, itertools.count.
22. Функции для обработки последовательностей: range, map, filter, sum, reduce, zip.
23. Объектно-ориентированное программирования (ООП) на языке программирования Python. Описание классов.
24. ООП в Python: атрибуты, свойства, сокрытие данных.
25. Типизация и полиморфизм.
26. Имитация типов. Перегрузка операторов. Магические методы.
27. ООП в Python. Особенности наследования.
28. Статические методы и методы класса.
29. Утиная типизация в Python. Менеджер контекста.
30. Работа с файлами в Python. Чтение бесконечных файлов.

31. Асинхронное программирование. Мотивация, проблемы и решения. Asyncio.async/await. Корутины.
32. GIL и многопоточные программы. Мьютексы и семафоры. Класс Thread. Поток, процесс.
33. Модули и пакеты.
34. GUI библиотеки. TKinter, PyQt,
35. Библиотеки numpy, matplotlib, pandas, sympy.
36. Фреймворки Django, aiohttp. Особенности применения.
37. Тестирование. Юнит и интеграционное тестирование.
38. Форматы CSV, json, xml. Чтение и запись средствами Python.
39. Регулярные выражения. Определение, шаблоны, основные методы.
40. Ускорение Python. Numba, Cython, интеграция с C/C++.

В филиале используется система с традиционной шкалой оценок – «отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «зачтено», «не зачтено» (далее – пятибалльная система).

Форма промежуточной аттестации по настоящей дисциплине – **экзамен**.

Применяемые критерии оценивания по дисциплинам (в соответствии с инструктивным письмом НИУ МЭИ от 14 мая 2012 года № И-23):

Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
«отлично»/ «зачтено (отлично)»/ «зачтено»	Выставляется обучающемуся, обнаружившему всестороннее, систематическое и глубокое знание материалов изученной дисциплины, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной рабочей программой дисциплины; проявившему творческие способности в понимании, изложении и использовании материалов изученной дисциплины, безупречно ответившему не только на вопросы билета, но и на дополнительные вопросы в рамках рабочей программы дисциплины, правильно выполнившему практическое задание. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «эталонный».
«хорошо»/ «зачтено (хорошо)»/ «зачтено»	Выставляется обучающемуся, обнаружившему полное знание материала изученной дисциплины, успешно выполняющему предусмотренные задания, усвоившему основную литературу, рекомендованную рабочей программой дисциплины; показавшему систематический характер знаний по дисциплине, ответившему на все вопросы билета, правильно выполнивший практическое задание, но допустивший при этом непринципиальные ошибки. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «продвинутый».
«удовлетворительно»/ «зачтено (удовлетворительно)»	Выставляется обучающемуся, обнаружившему знание материала изученной дисциплины в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, знакомому с основной литературой, рекомендованной рабочей программой дисциплины;

Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
«хорошо»/«зачтено»	допустившему погрешность в ответе на теоретические вопросы и/или при выполнении практических заданий, но обладающему необходимыми знаниями для их устранения под руководством преподавателя, либо неправильно выполнившему практическое задание, но по указанию преподавателя выполнившему другие практические задания из того же раздела дисциплины.. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «пороговый».
«неудовлетворительно»/не зачтено	Выставляется обучающемуся, обнаружившему серьезные пробелы в знаниях основного материала изученной дисциплины, допустившему принципиальные ошибки в выполнении заданий, не ответившему на все вопросы билета и дополнительные вопросы и неправильно выполнившему практическое задание (неправильное выполнение только практического задания не является однозначной причиной для выставления оценки «неудовлетворительно»). Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции на уровне «пороговый», закреплённые за дисциплиной, не сформированы.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Учебное и учебно-лабораторное оборудование

Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная:

– специализированной мебелью; доской аудиторной; демонстрационным оборудованием: персональным компьютером (ноутбуком); переносным (стационарным) проектором

Учебная аудитория для лабораторных работ, выполняемых в компьютерном классе, оснащенная:

– специализированной мебелью; доской аудиторной; персональными компьютерами с подключением к сети "Интернет" и доступом в ЭИОС филиала.

Для самостоятельной работы обучающихся по дисциплине используется помещение для самостоятельной работы обучающихся, оснащенное:

– специализированной мебелью; доской аудиторной; персональными компьютерами с подключением к сети "Интернет" и доступом в ЭИОС филиала.

Программное обеспечение

Операционная система OS Windows 10; Ubuntu 20+; офисный пакет Microsoft Office – для работы над РПД и методическим обеспечением к ней;

C++, Python, vscode – для решения технических задач разными методами программирования.

8. ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

для слепых и слабовидящих:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
- для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
- письменные задания оформляются увеличенным шрифтом;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

для глухих и слабослышащих:

- лекции оформляются в виде электронного документа;
- письменные задания выполняются на компьютере в письменной форме;
- экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

для лиц с нарушениями опорно-двигательного аппарата:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере;
- используется специальная учебная аудитория для лиц с ЛОВЗ – ауд. 106 главного учебного корпуса по адресу 214013, г. Смоленск, Энергетический пр-д, д.1, здание энергетического института (основной корпус).

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены филиалом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

для слепых и слабовидящих:

- в печатной форме увеличенным шрифтом;
- в форме электронного документа;
- в форме аудиофайла.

для глухих и слабослышащих:

- в печатной форме;
- в форме электронного документа.

для обучающихся с нарушениями опорно-двигательного аппарата:

- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

9. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература.

1. Мейер, Б. Основы программирования : [16+] / Б. Мейер. – 2-е изд., исправ. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 423 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=578061> .

2. Литвиненко, В.А. Программирование на С++ задач на графах : учебное пособие / В.А. Литвиненко ; Южный федеральный университет, Инженерно-технологическая академия. – Таганрог : Южный федеральный университет, 2016. – 83 с. : схем., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=493220> .

3. Шелудько, В.М. Основы программирования на языке высокого уровня Python – учебное пособие / В.М. Шелудько ; Министерство науки и высшего образования РФ, Южный федеральный университет, Инженерно-технологическая академия. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 147 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=500056> .

4. Лафоре Р. Объектно-ориентированное программирование в С++. – СПб.: Питер, 2007.

5. Павловская Т.А. С++. Объектно-ориентированное программирование: практикум : учебное пособие. – Санкт-Петербург : Питер, 2006 (и др. издания).

6. Дубов И.Р. Объектно-ориентированное программирование: учебное пособие. – Владимир: Владимирский государственный университет, 2003

7. Лаптев В.В., Морозов А.В., Бокова А.В. С++. Объектно-ориентированное программирование. Задачи и упражнения. – СПб.: Питер, 2007.

Дополнительная литература.

1. Нагаева, И.А. Основы алгоритмизации и программирования: практикум : [12+] / И.А. Нагаева, И.А. Кузнецов. – Москва ; Берлин : Директ-Медиа, 2021. – 169 с. : схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=598404>.

2. Приемы объектно-ориентированного проектирования. Паттерны проектирования: пер. с англ. / Э. Гамма [и др.] .– Санкт-Петербург : ДМК Пресс : Питер, 2008 (и др. издания).

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Номер изменения	Номера страниц				Всего страниц в документе	Наименование и № документа, вводящего изменения	Подпись, Ф.И.О. внесшего изменения в данный экземпляр	Дата внесения изменения в данный экземпляр	Дата введения изменения
	измененных	замененных	новых	аннулированных					
1	2	3	4	5	6	7	8	9	10