

ПЕРВОЕ ВЫСШЕЕ ТЕХНИЧЕСКОЕ УЧЕБНОЕ ЗАВЕДЕНИЕ РОССИИ



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОРНЫЙ УНИВЕРСИТЕТ

СОГЛАСОВАНО

УТВЕРЖДАЮ

Руководитель ОПОП ВО
доцент Ю.В. Ильюшин

Проректор по образовательной
деятельности
доцент Д.Г. Петраков

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ПРОГРАММИРОВАНИЕ И ОСНОВЫ АЛГОРИТМИЗАЦИИ СИСТЕМ
УПРАВЛЕНИЯ

Уровень высшего образования:	Бакалавриат
Направление подготовки:	27.03.04 Управление в технических системах
Направленность (профиль):	Информационные технологии в управлении
Квалификация выпускника:	бакалавр
Форма обучения:	очная
Составитель:	доцент Кухарова Т.В.

Санкт-Петербург

Рабочая программа дисциплины «Программирование и основы алгоритмизации систем управления» разработана:

– в соответствии с требованиями ФГОС ВО – бакалавриат по направлению подготовки «27.03.04 Управление в технических системах», утверждённого приказом Минобрнауки России №871 от 31 июля 2020 г.;

– на основании учебного плана бакалавриата по направлению подготовки «27.03.04 Управление в технических системах» направленность (профиль) «Информационные технологии в управлении».

Составитель _____ к.т.н., доцент Т.В. Кухарова

Рабочая программа рассмотрена и одобрена на заседании кафедры системного анализа и управления от «05» февраля 2021 г., протокол № 8.

Заведующий кафедрой _____ д.т.н., доц. Ю.В. Ильюшин

Рабочая программа согласована:

Начальник отдела лицензирования, аккредитации и контроля качества образования _____ Ю.А. Дубровская

Начальник отдела методического обеспечения учебного процесса _____ к.т.н. А.Ю. Романчиков

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Цель дисциплины «Программирование и основы алгоритмизации систем управления» — формирование у обучающихся знаний принципов разработки типовых алгоритмов, теории и технологии программирования и навыков разработки программного обеспечения для решения профессиональных задач.

Основными задачами дисциплины «Программирование и основы алгоритмизации систем управления» являются: поэтапное формирование у студентов знаний и умений применять типовые алгоритмы, основные концепции структурного и процедурного программирования, технологии объектно-ориентированного программирования, стандартные библиотеки языка, технологии проектирования программных продуктов с графическим интерфейсом пользователя для решения профессиональных задач.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Программирование и основы алгоритмизации систем управления» относится к обязательной части основной профессиональной образовательной программы по направлению подготовки «27.03.04 Управление в технических системах» и изучается в 3 и 4 семестрах.

Предшествующими курсами, на которых непосредственно базируется дисциплина «Программирование и основы алгоритмизации систем управления» являются «Информатика», «Математика», «Методы и средства обработки информации», «Информационные технологии в управлении техническими системами».

Дисциплина «Программирование и основы алгоритмизации систем управления» является основополагающей для изучения следующих дисциплин: «Программирование систем управления», «Алгоритмизация и управление техническими системами», «Телекоммуникационные системы управления», «Информационное обеспечение систем управления», «Системы управления базами данных».

Особенностью дисциплины является направленность на выработку навыков применения основных методов разработки прикладных программ, ориентированных на решение профессиональных задач.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины «Программирование и основы алгоритмизации систем управления» направлен на формирование следующих компетенций:

Формируемые компетенции		Код и наименование индикатора достижения компетенции
Содержание компетенции	Код компетенции	
Способен разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности	ОПК-6	ОПК-6.2. Знать логику построения и принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ
		ОПК-6.3. Знать современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий

Формируемые компетенции		Код и наименование индикатора достижения компетенции
Содержание компетенции	Код компетенции	
		<p>ОПК-6.4. Уметь выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач</p> <p>ОПК-6.5. Уметь применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий</p> <p>ОПК-6.6. Уметь читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения</p> <p>ОПК-6.7. Уметь анализировать профессиональные задачи, разрабатывать подходящие ИТ-решения</p> <p>ОПК-6.9. Владеть навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения</p> <p>ОПК-6.10. Владеть навыками отладки и тестирования прототипов программно-технических комплексов задач</p>
Способен выполнять эксперименты по заданным методикам и обрабатывать результаты с применением современных информационных технологий и технических средств	ОПК-9	<p>ОПК-9.1. Уметь: выполнять эксперименты по заданным методикам</p> <p>ОПК-9.2. Уметь: выполнять анализ полученных экспериментальных данных</p>
Способен использовать навыки анализа технологического оборудования, методы и средства технологического оснащения, средства измерения, приемы и методы работы с ними, применяемые при выполнении технологических процессов	ПКС-1	<p>ПКС-1.5. Владеть: навыками работы на компьютере, оснащенном специализированным программным обеспечением</p>

Формируемые компетенции		Код и наименование индикатора достижения компетенции
Содержание компетенции	Код компетенции	
Способен анализировать существующую структуру и методы оптимизации технологических и вспомогательных операций при проектировании устройств и систем автоматизации и управления	ПКС-2	ПКС-2.1. Знать: этапы и особенности проектирования блоков, элементов и систем автоматизации
		ПКС-2.3. Уметь: строить структурные схемы технологических процессов, проводить их расчет и оптимизацию

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Объем дисциплины и виды учебной работы

Общая трудоёмкость учебной дисциплины составляет 6 зачётных единиц, 216 ак. часов.

Вид учебной работы	Всего ак. часов	Ак. часы по семестрам	
		3	4
Аудиторная работа, в том числе:	85	34	51
Лекции (Л)	34	17	17
Практические занятия (ПЗ)	51	17	34
Лабораторные работы (ЛР)	-	-	-
Самостоятельная работа студентов (СРС), в том числе:	95	56	39
Выполнение курсовой работы (проекта)	20	-	20
Расчетно-графическая работа (РГР)	-	-	-
Реферат	-	-	-
Подготовка к практическим занятиям	73	56	17
Подготовка к лабораторным занятиям	-	-	-
Подготовка к зачету / дифф. зачету	2	-	2
Промежуточная аттестация – экзамен (Э), зачет (З), курсовая работа (КР)	36	Э (36)	З, КР
Общая трудоёмкость дисциплины			
ак. час.	216	126	90
зач. ед.	6	3.5	2.5

4.2. Содержание дисциплины

Учебным планом предусмотрены: лекции, практические занятия и самостоятельная работа.

4.2.1. Разделы дисциплины и виды занятий

Наименование разделов	Виды занятий				
	Всего ак. часов	Лекции	Практические занятия	Лабораторные работы	Самостоятельная работа студента, в том числе курсовая работа (проект)
Раздел 1 «Разработка приложений на языке С++»	92	17	19	-	56
Раздел 2 «Создание Python-приложений»	88	17	32	-	39
Итого:	180	34	51	-	95

4.2.2.Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание лекционных занятий	Трудоемкость в ак. часах
1	Раздел 1 «Разработка приложений на языке С++»	Парадигмы программирования. Процедурное программирование. Модульное программирование. Объектно-ориентированное программирование. Блок-схемы программ: символы данных и процессов, границы циклов, символы линий и специальные символы, правила применения символов и выполнения схем. Особенности языка программирования С++. Алфавит, литералы и комментарии. Типы данных языка С++. Операции языка С++. Операции преобразования типов в С++. Операторы языка С++. Структура программы на языке С++. Объявление и инициализация переменных и констант в С++. Объявление и определение функций в С++. Типы функций и семантика возврата значения из функции в С++. Директивы препроцессора в С++. Массивы в С++. Строки в С++. Поточный ввод/вывод в С++. Стандартные потоки для ввода и вывода. Файловые потоки. Указатели в С++. Операторы распределения памяти new и delete. Указатели и динамические массивы. Косвенная адресация. Структуры в С++: операции доступа к элементам структуры, вложенные структуры, инициализация структур. Классы в С++: спецификаторы доступа к членам класса, объявление класса, реализация класса.	17
2	Раздел 2 «Создание Python-приложений»	Особенности языка программирования Python. Целочисленная арифметика. Типы данных. Работа со строками в языке Python. Сложные структуры данных: списки, кортежи, словари и множества. Структура кода в языке Python. Линейные алгоритмы и алгоритмы ветвления. Работа с циклами в Python. Генерирование числовых последовательностей. Работа с функциями в Python.	17

№ п/п	Наименование раздела дисциплины	Содержание лекционных занятий	Трудоемкость в ак. часах
		Модули, программы и пакеты. Сложная обработка текстовых строк. Работа с файлами в языке Python. Объектно-ориентированное программирование в Python. Разработка приложений с графическим интерфейсом.	
Итого:			34

4.2.3. Практические занятия

№ п/п	Разделы	Тематика практических занятий	Трудоемкость в ак. часах
1	Раздел 1	Основы программирования на C++	2
2	Раздел 1	Разветвляющиеся процессы	2
3	Раздел 1	Циклические процессы	2
4	Раздел 1	Массивы	2
5	Раздел 1	Функции	2
6	Раздел 1	Указатели	2
7	Раздел 1	Файловый ввод/вывод данных	2
8	Раздел 1	Ввод/вывод данных через форму В C++ Builder	3
9	Раздел 1	Построение графиков в C++ Builder	2
10	Раздел 2	Введение в язык Python. Целочисленная арифметика. Типы данных	2
11	Раздел 2	Работа со строками в языке Python	2
12	Раздел 2	Сложные структуры данных: списки, кортежи	2
13	Раздел 2	Словари, множества в языке Python	2
14	Раздел 2	Структура кода в языке Python. Линейные алгоритмы и алгоритмы ветвления	2
15	Раздел 2	Работа с циклами в Python	2
16	Раздел 2	Генерирование числовых последовательностей	2
17	Раздел 2	Работа с функциями в Python	2
18	Раздел 2	Модули, программы и пакеты	2
19	Раздел 2	Работа с файлами в языке Python	2
20	Раздел 2	Моделирование с использованием модуля turtle	2
21	Раздел 2	Пользовательские подпрограммы и моделирование. Модуль Tkinter	2
22	Раздел 2	Моделирование физического явления	2
23	Раздел 2	Математика и статистика в стандартной библиотеке Python	2
24	Раздел 2	Модуль для научных расчетов NumPy. Работа с массивами	2
Итого:			51

4.2.4. Лабораторные работы

Лабораторные работы не предусмотрены

4.2.5. Курсовые работы

№ п/п	Темы курсовых работ
1.	Разработка приложения для расчета необходимого количества запасного имущества и приборов для устройств и систем
2.	Разработка приложения для расчета статистики дисперсионно-рангового критерия Краскела

	и Уоллиса
3.	Разработка приложения для расчёта площади фигур, заданных точками на плоскости
4.	Разработка программного продукта для формирования библиографических списков в соответствии с нормативными требованиями
5.	Разработка приложения для перевода чисел из одной системы счисления в другую
6.	Разработка приложения для вычисления центра тяжести плоской ограниченной фигуры с помощью двойного интеграла
7.	Разработка приложения для расчета траектории движения тела, брошенного под углом к горизонту
8.	Разработка приложения для осуществления тайм-менеджмента
9.	Разработка приложения для определения структуры затрат предприятия
10.	Разработка приложения, реализующего принятие решений в условиях неопределенности
11.	Разработка приложения для оценки устойчивости линейной системы по критерию Рауса
12.	Разработка приложения для решения линейных однородных дифференциальных уравнений второго порядка с постоянными коэффициентами
13.	Разработка приложения для нахождения экстремума функции двух переменных
14.	Разработка приложения для моделирования процесса распространения тепла в одномерном объекте
15.	Разработка приложения для расчета надежности восстанавливаемых резервированных систем
16.	Разработка системы автоматизации управления персоналом
17.	Разработка приложения для автоматизации расчета оптимальных количественных и качественных характеристик набора продуктов питания человека на месяц

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В ходе обучения применяются:

Лекции, которые являются одним из важнейших видов учебных занятий и составляют основу теоретической подготовки обучающихся. Цели лекционных занятий:

-дать систематизированные научные знания по дисциплине, акцентировать внимание на наиболее сложных вопросах дисциплины;

-стимулировать активную познавательную деятельность обучающихся, способствовать формированию их творческого мышления.

Практические занятия. Цели практических занятий:

-совершенствовать умения и навыки решения практических задач.

Главным содержанием этого вида учебных занятий является работа каждого обучающегося по овладению практическими умениями и навыками профессиональной деятельности.

Консультации (текущая консультация, накануне зачета и экзамена) является одной из форм руководства учебной работой обучающихся и оказания им помощи в самостоятельном изучении материала дисциплины, в ликвидации имеющихся пробелов в знаниях, задолженностей по текущим занятиям, в подготовке письменных работ (проектов).

Текущие консультации проводятся преподавателем, ведущим занятия в учебной группе, научным руководителем и носят как индивидуальный, так и групповой характер.

Самостоятельная работа обучающихся направлена на углубление и закрепление знаний, полученных на лекциях и других занятиях, выработку навыков самостоятельного активного приобретения новых, дополнительных знаний, подготовку к предстоящим учебным занятиям и промежуточному контролю.

Курсовая работа позволяет обучающимся развить навыки научного поиска.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

6.1. *Оценочные средства для самостоятельной работы и текущего контроля успеваемости*

Раздел 1. Разработка приложений на языке C++

1. Особенности языка программирования C++.
2. Алфавит, литералы и комментарии в C++.
3. Типы данных языка C++. Объявление typedef в C++.
4. Операции языка C++.
5. Операции преобразования типов в C++.
6. Операции составного присваивания в C++.
7. Условная операция в C++.
8. Порядок вычисления операндов в C++.
9. Операторы языка C++.
10. Оператор обработки исключений в C++.
11. Условный оператор в C++.
12. Оператор-переключатель в C++.
13. Оператор цикла с предусловием в C++.
14. Оператор цикла с постусловием в C++.
15. Оператор пошагового цикла в C++.
16. Операторы разрыва, продолжения и возврата в C++.
17. Структура программы на языке C++.
18. Объявление и инициализация переменных и констант в C++.
19. Объявление и определение функций в C++.
20. Типы функций и семантика возврата значения из функции в C++.
21. Встраиваемые функции в C++.
22. Задание параметров функций по умолчанию в C++.
23. Функции с переменным числом параметров в C++.
24. Перегрузка функций в C++.
25. Рекурсивные функции в C++.
26. Директивы препроцессора в C++.
27. Включение файлов и макроподстановки в C++.
28. Условная компиляция в C++.
29. Объявление и инициализация массива в C++.
30. Доступ к элементу массива и обработка массивов в C++.
31. Указатели в C++. Объявление и инициализация указателя.
32. Операции с указателями в C++.
33. Указатели и массивы в C++.
34. Операторы распределения памяти new и delete в C++.
35. Указатели и динамические массивы в C++.
36. Указатели и спецификатор const в C++.
37. Массивы указателей в C++.
38. Косвенная адресация в C++.
39. Структуры в C++. Операции доступа к элементам структуры.
40. Вложенные структуры в C++.
41. Инициализация структур в C++.
42. Массивы структур в C++.
43. Спецификаторы доступа к членам класса в C++.
44. Объявление класса в C++.
45. Реализация класса в C++.
46. Объявление объекта класса и доступ к членам объектов в C++.

47. Блок-схемы программ. Символы данных и процессов.
48. Блок-схемы программ. Границы циклов. Символы линий и специальные символы.
49. Блок-схемы программ. Правила применения символов и выполнения схем.
50. Строки в C++. Строковые литералы и массивы символов.
51. Поточковый ввод/вывод в C++. Стандартные потоки для ввода и вывода.
52. Поточковый ввод/вывод в C++. Файловые потоки.
53. Парадигмы программирования. Процедурное программирование.
54. Парадигмы программирования. Модульное программирование.
55. Парадигмы программирования. Объектно-ориентированное программирование.

Раздел 2. Создание Python-приложений

1. Каковы особенности синтаксиса языка Python?
2. Какие вычислительные операции применимы к целочисленным типам данных? К вещественным?
3. Что такое строковый тип? Какие операции применимы к строковому типу?
4. Каковы особенности ввода данных в Python?
5. Каковы особенности вывода данных в Python?
6. Что такое IDLE?
7. Какой командой определяется тип переменной?
8. Как применить операцию деления нацело?
9. Что произойдет при делении на 0 с помощью оператора `/`? С помощью `//`?
10. В чём особенность строк по сравнению с другими составными типами данных в Python и со строками во многих других языках программирования?
11. Какие способы задания строк вы знаете?
12. Для чего используются два вида кавычек при задании строк?
13. Как можно соединить 2 строки в одну?
14. Для чего предназначена функция `len()`?
15. Для чего предназначена функция `split()`?
16. Как изменится строка после применения функции `capitalize`?
17. Какая функция позволяет вывести все слова в строке с прописных букв?
18. Для чего используется функция `replace()`? Приведите пример её использования.
19. В чём отличие списков от строк?
20. Какие способы создания списков вы знаете?
21. Какая функция получает из строки список? Как задать разделитель, который должен быть использован?
22. Что будет, если при создании списка из строки разделитель встретился несколько раз подряд?
23. Как создать список списков? Как выводятся его элементы? Приведите пример.
24. Можно ли заменить элемент списка?
25. Как вывести диапазон элементов списка?
26. Как вывести на экран каждый второй элемент списка от последнего к первому?
27. Как вывести элементы списка на экран от последнего до первого?
28. Какие способы добавления элементов в список вы знаете?
29. Какие способы удаления элементов из списка вы знаете? В чём их отличие?
30. Как объединить два списка?
31. Что такое методы FIFO и LIFO?
32. Как определить номер элемента в списке по значению?
33. Как можно определить наличие элемента в списке?
34. Как можно определить количество вхождений элемента в список?
35. Как преобразовать список в строку?

36. В чём разница между функциями `sort` и `sorted`?
37. Каков порядок сортировки списка по умолчанию?
38. Как отсортировать список в обратном порядке?
39. Как определить длину списка?
40. В чём особенность присвоения значения одной переменной, содержащей список, другой?
41. Как можно скопировать список, сделав его независимым?
42. В чём состоит особенность кортежей?
43. Как создать кортеж? Какие способы вы знаете?
44. Какие достоинства и недостатки кортежей по сравнению со списками вы знаете?
45. В чём отличие словарей от других структур данных в Python?
46. Как создать словарь?
47. Как создать словарь из кортежа? Из списка? Из списка двухсимвольных строк?
48. Как добавить данные в словарь?
49. Какие требования предъявляются значению ключа в словаре?
50. Что произойдет, если новый ключ совпадет с уже имеющимся?
51. Как объединить два словаря?
52. Как удалить элементы из словаря?
53. Как очистить словарь целиком?
54. Как получить элемент словаря по ключу?
55. Какая функция ищет в словаре заданный элемент и возвращает заданное значение, если элемент не найден?
56. Как получить ключи словаря?
57. Как получить значения словаря?
58. Как скопировать значения из одного словаря в другой и избежать их связывания?
59. В чём особенность множеств?
60. Как создать пустое множество?
61. Что такое пересечение множеств? Как его найти? Какой будет получен тип данных?
62. Что такое объединение множеств? Как его найти?
63. Что такое разность множеств? Как ее получить?
64. Что такое «исключающее ИЛИ»? Как его получить?
65. Как проверить, является ли одно множество подмножеством другого?
66. Как создать кортеж из списков?
67. Как создать список списков?
68. Как создать словарь из списков?
69. Какие структуры можно использовать в качестве ключа словаря?
70. Как можно продлить строку? Связать несколько строк в одну?
71. Какие операторы сравнения в Python вы знаете?
72. Какова структура оператора `if`?
73. Как записать в условии знак равенства?
74. Как записать в условии знак неравенства?
75. Как задать условие «меньше либо равно», «больше либо равно»?
76. Как в условии можно проверить включение?
77. Что возвращает оператор сравнения?
78. Как и для чего используются булевы операторы `and`, `or`, `not`?
79. Какой приоритет имеют булевы операторы по сравнению с операторами сравнения?
80. Что произойдет, если элемент, который проверяется в условии, не будет булевого типа?
81. Для чего используются циклы?
82. Опишите особенности применения цикла `while`.

83. В чём разница между `elif` и `else`?
84. Для чего используется оператор `break`?
85. Для чего используется оператор `continue`?
86. Как можно проверить, завершился ли цикл досрочно?
87. В чём особенности синтаксиса цикла `for`?
88. Как выполнить итерирование по словарю? Как вывести ключи?
89. Как выполнить итерирование словаря и вывести его значения?
90. Как вывести и ключи и значения словаря?
91. Как используется прерывание в цикле `for`? Для чего?
92. Генерация и итерирование последовательностей.
93. Итерирование по нескольким последовательностям.
94. Генерирование числовых последовательностей с помощью функции `range()`.
95. Включения списков и словарей, множества и генератора.
96. Приведите примеры применения генерации последовательностей.
97. Определение и вызов функций.
98. Синтаксис функций.
99. Использование параметров функций.
100. Использование оператора `return`.
101. Использование значения `None`.
102. Позиционные аргументы.
103. Аргументы — ключевые слова.
104. Значение параметра по умолчанию.
105. Получение аргументов — ключевых слов с помощью `**`.
106. Понятие модуля.
107. Стандартные и пользовательские модули.
108. Подключение модуля из стандартной библиотеки.
109. Использование функций модуля.
110. Передача аргумента в функцию модуля.
111. Ошибка импорта. Использование псевдонимов.
112. Инструкция `from`.
113. Выбор имени и места расположения модуля.
114. Для чего используется ввод и вывод информации в файл, приведите примеры?
115. Открытие файла, аргументы функции открытия.
116. Запись в текстовый файл с помощью функции `write()`.
117. Считывание данных из текстового файла с помощью функций `read()`, `readline()` и `readlines()`.
118. Записываем данные в бинарный файл с помощью функции `write()`.
119. Чтение бинарных файлов с помощью функции `read()`.
120. Закрытие файлов автоматически с помощью ключевого слова `with`.
121. Изменение позиции с помощью функции `seek()`.
122. Назначение модуля `turtle`.
123. Метод `mainloop()`, вывод окна.
124. Команды перемещения пера модуля `turtle`.
125. Настройка параметров пера.
126. Черчение объектов в модуле `turtle`.
127. Рисование в окне модуля `turtle`.
128. Создание графиков функций.
129. Импорт библиотеки `tkinter`.
130. Создание главного окна.
131. Создание виджетов, установка их свойств, определение событий, определение обработчиков событий.
132. Расположение виджетов на главном окне, отображение главного окна.

133. Каково назначение модуля math? Его основные функции?
134. Функции округления, особенности их использования.
135. Константы модуля math.
136. Как осуществляется работа с комплексными числами в Python?
137. Использование модуля decimal.
138. Использование модуля fractions.
139. Какими командами создаются массивы? Какие атрибуты у этих команд?
140. Как изменить форму массива?
141. Как получить элемент массива?
142. Какие операции линейной алгебры реализует NumPy?

6.2. Оценочные средства для проведения промежуточной аттестации (зачета, экзамена)

6.2.1. Примерный перечень вопросов/заданий к зачету/экзамену (по дисциплине):

1. Парадигмы программирования. Процедурное программирование.
2. Парадигмы программирования. Модульное программирование.
3. Парадигмы программирования. Объектно-ориентированное программирование.
4. Блок-схемы программ. Символы данных и процессов.
5. Блок-схемы программ. Границы циклов. Символы линий и специальные символы.
6. Блок-схемы программ. Правила применения символов и выполнения схем.
7. Особенности языка программирования C++.
8. Алфавит, литералы и комментарии в C++.
9. Типы данных языка C++. Объявление typedef в C++.
10. Операции языка C++.
11. Операции преобразования типов в C++.
12. Операции составного присваивания в C++.
13. Условная операция в C++.
14. Порядок вычисления операндов в C++.
15. Операторы языка C++.
16. Оператор обработки исключений в C++.
17. Условный оператор в C++.
18. Оператор-переключатель в C++.
19. Оператор цикла с предусловием в C++.
20. Оператор цикла с постусловием в C++.
21. Оператор пошагового цикла в C++.
22. Операторы разрыва, продолжения и возврата в C++.
23. Структура программы на языке C++.
24. Объявление и инициализация переменных и констант в C++.
25. Объявление и определение функций в C++.
26. Типы функций и семантика возврата значения из функции в C++.
27. Встраиваемые функции в C++.
28. Задание параметров функций по умолчанию в C++.
29. Функции с переменным числом параметров в C++.
30. Перегрузка функций в C++.
31. Рекурсивные функции в C++.
32. Директивы препроцессора в C++.
33. Включение файлов и макроподстановки в C++.
34. Условная компиляция в C++.
35. Объявление и инициализация массива в C++.
36. Доступ к элементу массива и обработка массивов в C++.
37. Указатели в C++. Объявление и инициализация указателя.
38. Операции с указателями в C++.

39. Указатели и массивы в C++.
40. Операторы распределения памяти new и delete в C++.
41. Указатели и динамические массивы в C++.
42. Указатели и спецификатор const в C++.
43. Массивы указателей в C++.
44. Косвенная адресация в C++.
45. Структуры в C++. Операции доступа к элементам структуры.
46. Вложенные структуры в C++.
47. Инициализация структур в C++.
48. Массивы структур в C++.
49. Спецификаторы доступа к членам класса в C++.
50. Объявление класса в C++.
51. Реализация класса в C++.
52. Объявление объекта класса и доступ к членам объектов в C++.
53. Строки в C++. Строковые литералы и массивы символов.
54. Поточный ввод/вывод в C++. Стандартные потоки для ввода и вывода.
55. Поточный ввод/вывод в C++. Файловые потоки.
56. Особенности языка программирования Python. Целочисленная арифметика. Типы данных в Python.
57. Работа со строками в языке Python.
58. Сложные структуры данных: списки, кортежи.
59. Словари и множества в Python.
60. Структура кода в языке Python. Линейные алгоритмы и алгоритмы ветвления.
61. Работа с циклами в Python.
62. Генерирование числовых последовательностей в Python.
63. Работа с функциями в Python.
64. Работа с модулями, программами и пакетами в Python.
65. Сложная обработка текстовых строк в Python.
66. Работа с файлами в языке Python.
67. Объектно-ориентированное программирование в Python.
68. Разработка приложений с графическим интерфейсом в Python.

6.2.2. Примерные тестовые задания к экзамену

Вариант №1

№ п/п	Вопрос	Варианты ответа
1.	На каком из этапов устраняются синтаксические и логические ошибки в программе?	<ol style="list-style-type: none"> 1. анализ результатов. 2. алгоритмизация вычислительного процесса. 3. составление программы. 4. отладка программы.
2.	Алгоритм решения задачи – это:	<ol style="list-style-type: none"> 1. система точно сформулированных правил, определяющих процесс преобразования входной информации в выходную за конечное число шагов. 2. формализованное представление реального объекта, процесса или явления. 3. графическое изображение материального процесса. 4. модель, преобразующая входную информацию в выходную.

3.	Всякая программа, написанная на языке C++, состоит из:	<ol style="list-style-type: none"> нескольких функций. одного оператора. одной или нескольких функций. нескольких циклов.
4.	Имена, которыми в программе обозначаются величины и функции:	<ol style="list-style-type: none"> конструкторы. переменные. идентификаторы. структуры.
5.	В языке C переменные с именами ABC и авс обозначают:	<ol style="list-style-type: none"> регистр не имеет значения. одинаковые переменные. разные переменные. равнозначные переменные.
6.	Какой из вариантов оператора цикла выполняется бесконечно:	<ol style="list-style-type: none"> <code>int f=5; while(1) {f++;}</code> <code>int f=5; while(f>0) {f--;}</code> <code>int f=5; while(f<0) {f++;}</code> <code>int f=5; while(f!=3) {f--;}</code>
7.	Какого типа переменная объявлена? <code>int x;</code>	<ol style="list-style-type: none"> вещественного типа. целого типа. символьного типа. указатель на целое.
8.	В языке C логическая операция сравнения «равно» обозначается символами:	<ol style="list-style-type: none"> = == != =\=
9.	Разветвляющийся алгоритм с полной альтернативой реализуется на языке программирования C оператором:	<ol style="list-style-type: none"> <code>if ... else</code>. <code>if</code>. <code>switch ... case</code>. <code>while</code>.
10.	В языке C символьный массив, рассматриваемый как строка, обязательно заканчивается:	<ol style="list-style-type: none"> точкой с запятой. точкой. нулевым символом. единицей.
11.	Массив, объявленный в программе на языке C, как <code>int A[10]</code> , содержит элементы:	<ol style="list-style-type: none"> от <code>A[0]</code> до <code>A[10]</code>. от <code>A[1]</code> до <code>A[10]</code>. от <code>A[0]</code> до <code>A[9]</code>. от <code>A[1]</code> до <code>A[9]</code>.
12.	<code>double f(int a);</code> – это прототип функции _____ .	<ol style="list-style-type: none"> без параметров, возвращающей целое значение. без параметров, возвращающей вещественное значение. с вещественным параметром, возвращающей целое значение. с целым параметром, возвращающей вещественное значение.

13.	Какого типа переменная объявлена? <i>double *q;</i>	1. вещественного типа. 2. указатель на вещественное. 3. символьного типа. 4. целого типа.
14.	Что будет выведено на экран? <i>int *point;</i> <i>point=new(int);</i> <i>*point=8;</i> <i>cout<<*point;</i>	1. адрес области памяти, адресуемой указателем <i>point</i> 2. адрес области памяти, в которой хранится указатель <i>point</i> 3. 88 4. 8
15.	Структура данных, работающая по принципу «последним пришел – первым вышел»:	1. список. 2. очередь. 3. стек. 4. дерево.
16.	Переменная, в которой хранится адрес другой переменной, называется:	1. классом. 2. указателем. 3. оператором. 4. константой.
17.	Буферизация ввода-вывода _____ операции с файлом:	1. значительно ускоряет. 2. значительно замедляет. 3. планирует. 4. сжимает.
18.	Текстовые файлы относятся к файлам _____ доступа.	1. прямого. 2. двоичного. 3. текстового. 4. последовательного.
19.	Класс – это:	1. это структурированный тип, включающий в себя в качестве элементов данные различных типов и функции, применяемые по отношению к этим данным. 2. это структурированный тип, включающий в себя в качестве элементов однотипные данные и функции, применяемые по отношению к этим данным. 3. это расширение действия стандартных операций. 4. набор переменных разного типа.
20.	Деструктор предназначен для:	1. освобождения памяти при удалении объекта класса. 2. доступа к защищенным компонентам класса. 3. инициализации объекта класса. 4. обнуления значений членов класса.

Вариант №2

№ п/п	Вопрос	Варианты ответа
-------	--------	-----------------

1.	Программа – это:	<ol style="list-style-type: none"> 1. набор идентификаторов, связанный с областью памяти, отведенной для их хранения. 2. машинная реализация алгоритма решения задачи. 3. вид объекта, объединяющего данные и функции в одно целое. 4. структура алгоритма решения задачи.
2.	Алгоритм, содержащий одно или несколько логических условий и имеющий несколько ветвей вычислений:	<ol style="list-style-type: none"> 1. циклический алгоритм. 2. разветвляющийся алгоритм. 3. логический алгоритм. 4. условный алгоритм.
3.	В языке C++ логическая операция сравнения «не равно» обозначается символами:	<ol style="list-style-type: none"> 1. \neq 2. \neq 3. $!=$ 4. $<>$
4.	Идентификатор <u>не может</u> включать в себя:	<ol style="list-style-type: none"> 1. пробел. 2. знак подчеркивания. 3. цифры. 4. буквы.
5.	Что будет выведено на экран? <pre>for(int k=0;k<=5;k+=2) cout<<k;</pre>	<ol style="list-style-type: none"> 1. 012345 2. 024 3. 135 4. 12345
6.	Переменные в языке программирования C объявляются:	<ol style="list-style-type: none"> 1. обязательно в начале программы. 2. обязательно перед функцией <i>main()</i>. 3. обязательно в начале функции <i>main()</i>. 4. обязательно перед их использованием.
7.	Какого типа переменная объявлена? <pre>char s;</pre>	<ol style="list-style-type: none"> 1. вещественного типа. 2. целого типа. 3. символьного типа. 4. указатель на целое.
8.	Циклический алгоритм с предусловием реализуется на языке программирования C оператором:	<ol style="list-style-type: none"> 1. <i>do ... while.</i> 2. <i>while.</i> 3. <i>while ... do.</i> 4. <i>if ... else.</i>
9.	В языке C первый элемент массива имеет индекс:	<ol style="list-style-type: none"> 1. 0 2. 1 3. 2 4. 010
10.	Имя массива - это:	<ol style="list-style-type: none"> 1. указатель на его значение. 2. указатель на его начало. 3. указатель на длину массива. 4. количество элементов массива.
11.	Массив, объявленный в программе на языке C, как <i>int A[20]</i> , содержит элементы:	<ol style="list-style-type: none"> 1. от <i>A[1]</i> до <i>A[20]</i>. 2. от <i>A[0]</i> до <i>A[19]</i>. 3. от <i>A[0]</i> до <i>A[20]</i>. 4. от <i>A[1]</i> до <i>A[19]</i>.

12.	<i>void f(int a);</i> – это прототип функции _____ .	<ol style="list-style-type: none"> 1. с целым параметром, не возвращающей значение. 2. без параметров, возвращающей вещественное значение. 3. с вещественным параметром, возвращающей целое значение. 4. без параметров, возвращающей целое значение.
13.	Объявить строковую переменную <i>S</i> , состоящую из 6 букв?	<ol style="list-style-type: none"> 1. <i>char S(5);</i> 2. <i>char S[6];</i> 3. <i>char S[7];</i> 4. <i>double S[6];</i>
14.	Какого типа переменная объявлена? <i>int *q;</i>	<ol style="list-style-type: none"> 1. вещественного типа. 2. целого типа. 3. указатель на целое. 4. символьного типа.
15.	Что будет выведено на экран? <i>int *point;</i> <i>point=new(int);</i> <i>*point=3;</i> <i>cout<<*point;</i>	<ol style="list-style-type: none"> 1. 3 2. *3 3. 33 4. адрес области памяти, адресуемой указателем <i>point</i>
16.	Структура данных, работающая по принципу «первым пришел – первым вышел»:	<ol style="list-style-type: none"> 1. дерево. 2. стек. 3. список. 4. очередь.
17.	Указатель – это:	<ol style="list-style-type: none"> 1. переменная, которая хранит свое значение. 2. операция, которая выделяет память для динамической переменной. 3. переменная, которая хранит адрес другой переменной. 4. переменная, которая хранит свой адрес.
18.	Двоичные файлы относятся к файлам _____ доступа.	<ol style="list-style-type: none"> 1. прямого. 2. двоичного. 3. текстового. 4. последовательного.
19.	Принадлежность функции к какому-либо классу обозначается знаком:	<ol style="list-style-type: none"> 1. ; 2. ! 3. : 4. ::
20.	Имя деструктора:	<ol style="list-style-type: none"> 1. должно начинаться с символа ‘&’, за которым следует имя класса. 2. должно начинаться с символа тильда ‘~’, за которым следует имя класса. 3. должно совпадать с именем класса. 4. может быть любым.

Вариант №3

№ п/п	Вопрос	Варианты ответа
-------	--------	-----------------

1.	_____ переводит компьютеру все инструкции программы непосредственно в момент их выполнения.	<ul style="list-style-type: none"> 1. компилятор. 2. интерпретатор. 3. диспетчер задач. 4. конструктор.
2.	В блок-схеме алгоритма условие обозначается:	<ul style="list-style-type: none"> 1. квадратом. 2. прямоугольником. 3. ромбом. 4. кругом.
3.	Величина изменения параметра цикла после каждого выполнения тела цикла называется:	<ul style="list-style-type: none"> 1. параметром. 2. метром. 3. скачком. 4. шагом.
4.	Элементы одного и того же типа, расположенные последовательно друг за другом в памяти, называются:	<ul style="list-style-type: none"> 1. структурой. 2. конструктором. 3. оператором. 4. массивом.
5.	Что будет выведено на экран? <pre>for(int k=0;k<3;k++) cout<<k;</pre>	<ul style="list-style-type: none"> 1. 0123 2. 024 3. 012 4. 12
6.	Разветвляющийся алгоритм с множественным выбором реализуется на языке программирования C оператором:	<ul style="list-style-type: none"> 1. <i>if ... else.</i> 2. <i>if.</i> 3. <i>switch ... case.</i> 4. <i>while.</i>
7.	Какого типа переменная объявлена? <pre>double y;</pre>	<ul style="list-style-type: none"> 1. вещественного типа. 2. целого типа. 3. символьного типа. 4. указатель на целое.
8.	Правильное объявление строки:	<ul style="list-style-type: none"> 1. <i>char string[4] = "ABCD";</i> 2. <i>char string[3] = "ABCD";</i> 3. <i>char string[5] = "ABCD";</i> 4. <i>char string[1] = "ABCD";</i>
9.	Что будет выведено на экран? <pre>char str[5]; str[0]='x'; str[1]='л'; str[2]='е'; str[3]='б'; str[4]='\0'; cout<<str;</pre>	<ul style="list-style-type: none"> 1. хлеб\0 2. хлеб0 3. хлеб 4. 'х' 'л' 'е' 'б' '\0'
10.	Какое значение имеет элемент массива с индексом 1, объявленного как <pre>int arr[5]={4,8,9,7,2};</pre>	<ul style="list-style-type: none"> 1. 4 2. 8 3. 9 4. 2

11.	Выберите прототип функции f , не возвращающей значение.	<ol style="list-style-type: none"> 1. <code>void f(int a);</code> 2. <code>int f(void);</code> 3. <code>double f(int a);</code> 4. <code>char f(void);</code>
12.	Какого типа переменная объявлена? <code>int *p;</code>	<ol style="list-style-type: none"> 1. вещественного типа. 2. указатель на целое.. 3. символьного типа. 4. целого типа
13.	Что будет выведено на экран? <code>int *point;</code> <code>point=new(int);</code> <code>*point=5;</code> <code>cout<<*point;</code>	<ol style="list-style-type: none"> 1. 5 2. *5 3. 55 4. адрес области памяти, адресуемой указателем <code>point</code>
14.	Для выделения памяти для динамической переменной используется оператор:	<ol style="list-style-type: none"> 1. <code>do</code> 2. <code>new</code> 3. <code>for</code> 4. <code>delete</code>
15.	Динамические данные – это:	<ol style="list-style-type: none"> 1. данные, память под которые может быть выделена и освобождена в течение выполнения программы. 2. данные, под которые не выделяется память. 3. данные, под которые выделяется память только при завершении программы. 4. данные, под которые выделяется память при запуске программы.
16.	Именованная область памяти на диске – это:	<ol style="list-style-type: none"> 1. алгоритм. 2. файл. 3. конструктор. 4. массив.
17.	_____ задает место в файле, откуда будет вводиться информация:	<ol style="list-style-type: none"> 1. фиксатор. 2. указатель считывания. 3. стабилизатор. 4. номер файла.
18.	Для работы с файлом необходимо сначала:	<ol style="list-style-type: none"> 1. записать данные в файл. 2. объявить файловую переменную. 3. прочитать данные из файла. 4. открыть файл только для чтения.
19.	Режим доступа к членам класса определяется:	<ol style="list-style-type: none"> 1. номером члена в классе. 2. именем конструктора. 3. спецификаторами доступа. 4. наличием деструктора.
20.	Имя конструктора:	<ol style="list-style-type: none"> 1. всегда начинается с символа ‘&’, за которым следует имя класса. 2. всегда начинается с символа тильда ‘~’, за которым следует имя класса. 3. должно совпадать с именем класса. 4. не должно совпадать с именем класса.

6.3. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

6.3.1. Критерии оценок промежуточной аттестации (экзамен)

Оценка			
«2» (неудовлетворительно)	Пороговый уровень освоения «3» (удовлетворительно)	Углубленный уровень освоения «4» (хорошо)	Продвинутый уровень освоения «5» (отлично)
Студент не знает значительной части материала, допускает существенные ошибки в ответах на вопросы	Студент поверхностно знает материал основных разделов и тем учебной дисциплины, допускает неточности в ответе на вопрос	Студент хорошо знает материал, грамотно и по существу излагает его, допуская некоторые неточности в ответе на вопрос.	Студент в полном объеме знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос
Не умеет находить решения большинства предусмотренных программой обучения заданий	Иногда находит решения, предусмотренные программой обучения заданий	Уверенно находит решения, предусмотренные программой обучения заданий	Безошибочно находит решения, предусмотренные программой обучения заданий
Большинство предусмотренных программой обучения заданий не выполнено	Предусмотренные программой обучения задания выполнены удовлетворительно	Предусмотренные программой обучения задания успешно выполнены	Предусмотренные программой обучения задания успешно выполнены

Примерная шкала оценивания знаний в тестовой форме:

Количество правильных ответов, %	Оценка
0-49	Неудовлетворительно
50-65	Удовлетворительно
66-85	Хорошо
86-100	Отлично

6.3.3. Критерии оценок промежуточной аттестации (зачет)

Оценка	Описание
Зачтено	Посещение более 50 % лекционных и практических занятий; студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос; все предусмотренные программой обучения задания выполнены, качество их выполнения достаточно высокое; в течение семестра выполнил творческую работу.
Не зачтено	Посещение менее 50 % лекционных и практических занятий; студент не знает значительной части материала, допускает существенные ошибки в ответах на вопросы; большинство предусмотренных программой обучения заданий не выполнено, качество их выполнения оценено числом баллов, близким к минимальному.

Примерная шкала оценивания знаний в тестовой форме:

Количество правильных ответов, %	Оценка
0-49	Не зачтено
50-65	Зачтено
66-85	Зачтено
86-100	Зачтено

6.3.4. Описание шкалы и критериев оценивания для проведения промежуточной аттестации в форме защиты курсовой работы

Студент выполняет курсовую работу в соответствии с графиком, принятым на заседании кафедры. Оценка может быть снижена за несоблюдение установленного кафедрой графика.

Оценка			
«2» (неудовлетворительно)	Пороговый уровень освоения	Углубленный уровень освоения	Продвинутый уровень освоения
	«3» (удовлетворительно)	«4» (хорошо)	«5» (отлично)
Студент не выполнил курсовую работу в соответствии с заданием. Не владеет теоретическими знаниями по изучаемой дисциплине. Необходимые практические компетенции не сформированы	Студент выполнил курсовую работу с существенными ошибками. При защите курсового проекта демонстрирует слабую теоретическую подготовку. При решении задач, предусмотренных программой учебной дисциплины, допускает неточности, существенные ошибки	Студент выполнил курсовую работу с некоторыми незначительными ошибками и неточностями. При защите курсового проекта демонстрирует хорошую теоретическую подготовку. Хорошо справляется с решением задач, предусмотренных программой учебной дисциплины	Студент выполнил курсовую работу полностью в соответствии с заданием. При защите курсового проекта демонстрирует высокую теоретическую подготовку. Успешно справляется с решением задач, предусмотренных программой учебной дисциплины

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Рекомендуемая литература

7.1.1. Основная литература

1. Златопольский, Д. М. Программирование: типовые задачи, алгоритмы, методы : учебное пособие / Д. М. Златопольский. — 4-е изд., электрон. — Москва : Лаборатория знаний, 2020. — 226 с. — ISBN 978-5-00101-789-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1094359> (дата обращения: 30.09.2021). – Режим доступа: по подписке.

2. Затонский, А. В. Программирование и основы алгоритмизации. Теоретические основы и примеры реализации численных методов: учебное пособие / А.В. Затонский, Н.В. Бильфельд. — 2-е изд. — Москва : РИОР : ИНФРА-М, 2020. — 167 с. — (Высшее образование). — DOI: <https://www.dx.doi.org/10.12737/20468>. - ISBN 978-5-369-01195-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1077389> (дата обращения: 30.09.2021). – Режим доступа: по подписке.

3. Окулов, С. М. Программирование в алгоритмах / Окулов С.М., - 6-е изд., (эл.) - Москва :Лаборатория знаний, 2017. - 386 с.: ISBN 978-5-00101-449-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/502153> (дата обращения: 30.09.2021). – Режим доступа: по подписке.

7.1.2. Дополнительная литература

1. Литвиненко, В. А. Программирование на C++ задач на графах: Учебное пособие / Литвиненко В.А. - Таганрог:Южный федеральный университет, 2016. - 83 с.: ISBN 978-5-9275-2311-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/997083> (дата обращения: 30.09.2021). – Режим доступа: по подписке.

2. Златопольский, Д.М. Основы программирования на языке Python / Д.М. Златопольский. - Москва : ДМК Пресс, 2017. - 284 с. - ISBN 978-5-97060-552-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1028147> (дата обращения: 30.09.2021). – Режим доступа: по подписке.

3. Жуков, Р. А. Язык программирования Python: практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1412168> (дата обращения: 30.09.2021). – Режим доступа: по подписке.

7.1.3. Учебно-методическое обеспечение

1. Новожилов И.М. Учебно-методические материалы для проведения практических занятий по учебной дисциплине «Программирование и основы алгоритмизации систем управления». Режим доступа: <http://ior.spmi.ru/>.

2. Новожилов И.М. Методические рекомендации по выполнению курсовой работы по учебной дисциплине «Программирование и основы алгоритмизации систем управления». Режим доступа: <http://ior.spmi.ru/>.

3. Новожилов И.М. Учебно-методические материалы для проведения самостоятельной работы по учебной дисциплине «Программирование и основы алгоритмизации систем управления». Режим доступа: <http://ior.spmi.ru/>.

7.2. Базы данных, электронно-библиотечные системы, информационно-справочные и поисковые системы

1. Сайт Российской государственной библиотеки: <http://www.rsl.ru>.

2. Сайт Государственной публичной научно-технической библиотеки России: <http://www.gpntb.ru>.

3. Электронно-библиотечная система «Лань»: <https://e.lanbook.com/books>.

4. Научная электронная библиотека «eLIBRARY»: <https://elibrary.ru>.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Материально-техническое оснащение аудиторий:

1. Аудитория для проведения лекционных, практических занятий и лабораторных работ
Оснащенность помещения: 16 посадочных мест. Стол аудиторный – 10 шт., компьютерное кресло – 23 шт., моноблок – 17 шт. (возможность доступа к сети «Интернет»), доска аудиторная под фло-мастер – 1 шт., лазерный принтер – 1 шт.

Перечень лицензионного программного обеспечения: Microsoft Windows 7 Professional (ГК № 1464-12/10 от 15.12.10) Microsoft Office 2007 Professional Plus (Microsoft Open License 46082032 от 30.10.2009, GPSS World (свободно распространяемое ПО), Arduino Software (IDE) (свободно распространяемое ПО), Microsoft SQL Server Express (свободно распространяемое ПО).

2. Аудитория для проведения лекционных, практических занятий и лабораторных работ
Оснащенность помещения: 16 посадочных мест. Стол аудиторный – 9 шт., компьютерное кресло – 17 шт., моноблок – 17 шт. (возможность доступа к сети «Интернет»), лазерный принтер – 1 шт., доска – 1 шт.

Перечень лицензионного программного обеспечения: Microsoft Windows 7 Professional (ГК № 1464-12/10 от 15.12.10) Microsoft Office 2007 Professional Plus (Microsoft Open License 46082032 от 30.10.2009 MathCad Education (Договор №1134-11/12 от 28.11.2012), GPSS World (свободно распространяемое ПО), Arduino Software (IDE) (свободно распространяемое ПО), Microsoft SQL Server Express (свободно распространяемое ПО).

8.2. Помещения для самостоятельной работы:

1. Оснащенность помещения для самостоятельной работы: 13 посадочных мест. Стул – 25 шт., стол – 2 шт., стол компьютерный – 13 шт., шкаф – 2 шт., доска аудиторная маркерная – 1 шт., АРМ учебное ПК (монитор + системный блок) – 14 шт. Доступ к сети «Интернет», в электронную информационно-образовательную среду Университета.

Перечень лицензионного программного обеспечения: Microsoft Windows 7 Professional:ГК № 1464-12/10 от 15.12.10 «На поставку компьютерного оборудования» ГК № 959-09/10 от 22.09.10

«На поставку компьютерной техники» ГК № 447-06/11 от 06.06.11 «На поставку оборудования» ГК № 984-12/11 от 14.12.11 «На поставку оборудования» Договор № 1105-12/11 от 28.12.2011 «На поставку компьютерного оборудования», Договор № 1106-12/11 от 28.12.2011 «На поставку компьютерного оборудования» ГК № 671-08/12 от 20.08.2012 «На поставку продукции», Microsoft Open License 60799400 от 20.08.2012, Microsoft Open License 48358058 от 11.04.2011, Microsoft Open License 49487710 от 20.12.2011, Microsoft Open License 49379550 от 29.11.2011, Microsoft Office 2010 Standard: Microsoft Open License 60799400 от 20.08.2012, Microsoft Open License 60853086 от 31.08.2012 Kaspersky antivirus 6.0.4.142.

2. Оснащенность помещения для самостоятельной работы: 17 посадочных мест. Доска для письма маркером – 1 шт., рабочие места студентов, оборудованные ПК с доступом в сеть университета – 17 шт., мультимедийный проектор – 1 шт., АРМ преподавателя для работы с мультимедиа – 1 шт. (системный блок, мониторы – 2 шт.), стол – 18 шт., стул – 18 шт. Доступ к сети «Интернет», в электронную информационно-образовательную среду Университета.

Перечень лицензионного программного обеспечения: Операционная система Microsoft Windows XP Professional: Microsoft Open License 16020041 от 23.01.200. Операционная система Microsoft Windows 7 Professional Microsoft Open License 49379550 от 29.11.2011. Microsoft Office 2007 Standard Microsoft Open License 42620959 от 20.08.2007.

3. Оснащенность помещения для самостоятельной работы: 16 посадочных мест. Стол компьютерный для студентов (тип 4) - 3 шт., стол компьютерный для студентов (тип 6) – 2 шт., стол компьютерный для студентов (тип 7) – 1 шт., кресло преподавателя (сетка, цвет черный) – 17 шт., доска напольная мобильная белая магнитно-маркерная «Magnetoplan» 1800мм×1200мм - 1 шт., моноблок Lenovo M93Z Intel Q87 – 17 шт., плакат – 5 шт. Доступ к сети «Интернет», в электронную информационно-образовательную среду Университета.

Перечень лицензионного программного обеспечения: Microsoft Windows 7 Professional: Microsoft Open License 49379550 от 29.11.2011. Microsoft Office 2007 Professional Plus: Microsoft Open License 46431107 от 22.01.2010. CorelDRAW Graphics Suite X5 Договор №559-06/10 от 15.06.2010 «На поставку программного обеспечения». Autodesk product: Building Design Suite Ultimate 2016, product Key: 766H1. Cisco Packet Tracer 7.1 (свободно распространяемое ПО), Quantum GIS (свободно распространяемое ПО), Python (свободно распространяемое ПО), R (свободно распространяемое ПО), Rstudio (свободно распространяемое ПО), SMath Studio (свободно распространяемое ПО), GNU Octave (свободно распространяемое ПО), Scilab (свободно распространяемое ПО).

8.3. Помещения для хранения и профилактического обслуживания оборудования:

1. Центр новых информационных технологий и средств обучения:

Оснащенность: персональный компьютер – 2 шт. (доступ к сети «Интернет»), монитор – 4 шт., сетевой накопитель – 1 шт. источник бесперебойного питания – 2 шт., телевизор плазменный Panasonic – 1 шт., точка Wi-Fi – 1 шт., паяльная станция – 2 шт., дрель – 5 шт., перфоратор – 3 шт., набор инструмента – 4 шт., тестер компьютерной сети – 3 шт., баллон со сжатым газом – 1 шт., паста теплопроводная – 1 шт., пылесос – 1 шт., радиостанция – 2 шт., стол – 4 шт., тумба на колесиках – 1 шт., подставка на колесиках – 1 шт., шкаф – 5 шт., кресло – 2 шт., лестница Alve – 1 шт.

Перечень лицензионного программного обеспечения: Microsoft Windows 7 Professional (Лицензионное соглашение Microsoft Open License 60799400 от 20.08.2012)

Microsoft Office 2010 Professional Plus (Лицензионное соглашение Microsoft Open License 60799400 от 20.08.2012)

Антивирусное программное обеспечение Kaspersky Endpoint Security (Договор № Д810(223)-12/17 от 11.12.17)

2. Центр новых информационных технологий и средств обучения:

Оснащенность: стол – 5 шт., стул – 2 шт., кресло – 2 шт., шкаф – 2 шт., персональный компьютер – 2 шт. (доступ к сети «Интернет»), монитор – 2 шт., МФУ – 1 шт., тестер компьютерной сети – 1 шт., балон со сжатым газом – 1 шт., шуруповерт – 1 шт.

Перечень лицензионного программного обеспечения: Microsoft Windows 7 Professional (Лицензионное соглашение Microsoft Open License 60799400 от 20.08.2012)

Microsoft Office 2007 Professional Plus (Лицензионное соглашение Microsoft Open License 46431107 от 22.01.2010)

Антивирусное программное обеспечение Kaspersky Endpoint Security (Договор № Д810(223)-12/17 от 11.12.17)

3. Центр новых информационных технологий и средств обучения:

Оснащенность: стол – 2 шт., стуля – 4 шт., кресло – 1 шт., шкаф – 2 шт., персональный компьютер – 1 шт. (доступ к сети «Интернет»), веб-камера Logitech HD C510 – 1 шт., колонки Logitech – 1 шт., тестер компьютерной сети – 1 шт., дрель – 1 шт., телефон – 1 шт., набор ручных инструментов – 1 шт.

Перечень лицензионного программного обеспечения: Microsoft Windows 7 Professional (Лицензионное соглашение Microsoft Open License 48358058 от 11.04.2011)

Microsoft Office 2007 Professional Plus (Лицензионное соглашение Microsoft Open License 46431107 от 22.01.2010)

Антивирусное программное обеспечение Kaspersky Endpoint Security (Договор № Д810(223)-12/17 от 11.12.17)

8.4. Лицензионное программное обеспечение:

1. Microsoft Windows 8 Professional (договор бессрочный ГК № 875-09/13 от 30.09.2013 «На поставку компьютерной техники»).

2. Microsoft Office 2007 Standard (договор бессрочный Microsoft Open License 42620959 от 20.08.2007).

3. Microsoft Office 2010 Professional Plus (договор бессрочный Microsoft Open License 60799400 от 20.08.2012, договор бессрочный Microsoft Open License 47665577 от 10.11.2010, договор бессрочный Microsoft Open License 49379550 от 29.11.2011).