

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
профессор Д.А. Первухин**

**Проректор по образовательной
деятельности Д.Г. Петраков**

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

Уровень высшего образования:	Магистратура
Направление подготовки:	27.04.04 Управление в технических системах
Направленность (профиль):	Анализ и синтез технических систем с распределенными параметрами
Квалификация выпускника:	магистр
Форма обучения:	очная
Составитель:	доцент Ильюшин Ю.В.

Рабочая программа дисциплины «Прикладное программирование» разработана:

– в соответствии с требованиями ФГОС ВО по направлению подготовки 27.04.04 «Управление в технических системах» (уровень магистратура), утвержденный. Приказом Минобрнауки России № 942 от 11 августа 2020 г.;

– на основании учебного плана направления «27.04.04 Управление в технических системах» направленность (профиль) «Анализ и синтез технических систем с распределёнными параметрами».

Составитель

д.т.н., доцент Ильюшин Ю.В.

Рабочая программа рассмотрена и одобрена на заседании кафедры системного анализа и управления от 01.02.2023 г., протокол № 10.

Заведующий кафедрой

профессор,
д.т.н.

Первухин Д.А.

1. ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель изучения дисциплины «Прикладное программирование» состоит в ознакомлении студентов с возможностями программирования задач, работающих с базами данных, приобретение знаний в области прикладного программирования и навыков в применении этих знаний для решения практических задач, необходимых для профессиональной деятельности.

«Прикладное программирование» изучается в качестве дисциплины профессионального цикла, необходимой для последующего логического перехода к изучению цикла профессиональных дисциплин по направлению 27.04.04 Управление в технических системах профиля подготовки: «Анализ и синтез технических систем с распределенными параметрами».

Задачи изучения дисциплины – усвоение теоретических основ современных технологий программирования, получение практической подготовки в области выбора и применения технологии программирования для задач автоматизации обработки информации и управления.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Прикладное программирование» относится к обязательной части Блока 1 «Дисциплины (модули)» основной профессиональной образовательной программы по направлению подготовки 27.04.04 «Управление в технических системах» (уровень магистратуры) и изучается в 1-м семестре.

Предшествующими курсами, на которых непосредственно базируется дисциплина «Прикладное программирование» являются «Информатика», «Физика», «Теория управления» и др. читаемые в курсе бакалавриата.

Дисциплина «Прикладное программирование» является основополагающей для изучения следующих дисциплин: «Компьютерные технологии управления в технических системах», «Автоматизированное проектирование средств и систем управления».

Особенностью дисциплины является ознакомлении студентов с возможностями программирования задач, работающих с базами данных, приобретение знаний в области прикладного программирования и навыков в применении этих знаний для решения практических задач, необходимых для профессиональной деятельности.

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

Процесс изучения дисциплины направлен на формирование следующих результатов обучения, представленных в таблице:

Формируемые компетенции		Код и наименование индикатора достижения компетенции
Содержание компетенции	Код компетенции	
Способен анализировать и выявлять естественно-научную сущность проблем управления в технических системах на основе положений, законов и методов в области	ОПК-1	ОПК-1.2 Уметь: определять подзадачи и надзадачи
		ОПК-1.3 Уметь: формировать комплексную задачу управления

естественных наук и математики		
Способен самостоятельно решать задачи управления в технических системах на базе последних достижений науки и техники	ОПК-3	ОПК-3.2 Владеть: навыками поиска и анализа предметной области с целью поиска оптимального управленческого решения
Способен анализировать комплексы программно-аппаратных средств управления, мониторинга и диагностики, приемами и методами математического и компьютерного моделирования, технологического процесса предприятия	ПКС-1	ПКС-1.1 Уметь: определять математические, технические и информационные связи между техническим оснащением, автоматизированными рабочими местами и подразделениями организации
		ПКС-1.2 Владеть: навыками анализа и обработки данных, методами построения математических, информационных и технических моделей
Способен выявлять трудоемкие операции	ПКС-3	ПКС-3.3. Уметь: разрабатывать специализированное программное обеспечение, обеспечивающее анализ структуры производственного процесса
Способен владеть навыками анализа, разработки, моделирования и внедрения элементов и систем автоматизации производственных процессов	ПКС-4	ПКС-4.1 Знать: технические характеристики и функциональные возможности программных средств автоматизации
		ПКС- 4.3. Уметь: проводить математическое и компьютерное моделирование на основе результатов экспериментальных и аналитических исследований, в том числе с разработкой специализированного программного обеспечения
		ПКС- 4.6. Владеть: техническими средствами автоматизации, включающими элементы управления, системы и средства измерения, а также информационное обеспечение указанных средств

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 академических часов.

Вид учебной работы	Всего часов	Часы по семестрам
		1
Аудиторные занятия (всего), в том числе:	56	56
Лекции	28	28
Практические занятия (ПЗ)	28	28
Самостоятельная работа студентов (СРС)	52	52
Подготовка к практическим занятиям	52	52
Вид аттестации (курсовой проект – КР; зачет – З)	3	3
Общая трудоемкость дисциплины		
час.	108	108

4.2 Содержание дисциплины

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

4.2.2. Разделы дисциплин и виды занятий:

№ п/п	Наименование разделов	Виды занятий				
		Всего ак. часов	Лекции	Практические занятия	Лабораторные работы	Самостоятельная работа студента, в том числе курсовая работа (проект)а
1.	Раздел 1 Проблемы разработки сложных программных систем. Унифицированный процесс разработки и экстремальное программирование. Анализ предметной области и требования к ПО. Качество ПО и методы его контроля Архитектур а программного обеспечения	44	14	10	-	20
2.	Раздел 2 Образцы проектирования Принципы создания удобного пользовательского интерфейса Разработка различных уровней Web-приложений в J2EE	64	14	18	-	32
Итого:		108	28	28	-	52

4.2.1. Содержание разделов дисциплины:

№ п/п	Разделы	Содержание лекционных занятий	Трудоемкость в ак. часах
1.	Раздел 1. Проблемы разработки сложных программных систем. Унифицированный процесс разработки и экстремальное программирование. Анализ предметной области и требования к ПО. Качество ПО и методы его контроля Архитектур а программного обеспечения	Жизненный цикл и процессы разработки ПО Программы «большие» и «маленькие» Принципы работы со сложными системами. Стандарты жизненного цикла «Тяжелые» и «легкие» процессы разработки. Унифицированный процесс Rational Качество ПО и методы его контроля Архитектур а программного обеспечения	14
2.	Раздел 2. Образцы проектирования Принципы создания удобного пользовательского интерфейса Разработка различных	Образцы проектирования Компонентные технологии и разработка распределенного ПО. Компонентные технологии разработки Web-приложений Развитие компонентных технологий Управление разработкой ПО	14

	уровней Web-приложений в J2EE		
Итого:			28

4.2.3 Лабораторный практикум

Лабораторные занятия не предусмотрены.

4.2.4 Практические занятия

№ п/п	№ раздела	Тематика практических занятий	Трудоемкость в ак. часах
1.	Раздел 1	Интегрированная среда разработки Delphi	10
2.	Раздел 2	Структура программы и типы в языке Delphi	18
Итого:			28

4.2.5 Примерные темы для проведения курсовых проектов

№ п/п	Темы курсовых работ / проектов
1	«Моделирование информационной системы измерения температурного поля печи индукционного нагрева».
2	«Моделирование системы управления температурным полем печи с импульсным источником нагрева».
3	«Моделирование системы управления температурным полем печи с импульсным источником нагрева».
4	«Моделирование информационной системы измерения температурного поля восходящего».
5	«Моделирование информационной системы измерения температурного поля печи с импульсным источником нагрева».
6	«Моделирование информационной системы измерения температурного поля пластины».
7	«Моделирование информационной системы измерения температурного поля печи индукционного нагрева».
8	«Моделирование информационной системы измерения температурного поля многослойной пластины».
9	«Моделирование информационной системы измерения температурного поля цилиндра».
10	«Моделирование информационной системы измерения температурного поля сферы».
11	«Моделирование информационной системы измерения температурного поля куба».
12	«Моделирование информационной системы измерения температурного поля изотропного цилиндра».

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.

Семинарские занятия. Цели семинарских занятий:

-углубить и закрепить знания, полученные на лекциях и в процессе самостоятельной работы обучающихся с учебной и научной литературой;

-обеспечить живое, творческое обсуждение учебного материала в форме дискуссии, обмена мнениями по рассматриваемым вопросам.

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) 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 Организация самостоятельной работы студентов

**Раздел 1. Проблемы разработки сложных программных систем. Унифицированный процесс разработки и экстремальное программирование. Анализ предметной области и требования к ПО. Качество ПО и методы его контроля
Архитектур а программного обеспечения**

1. Управление разработкой ПО.
2. Организационная культура.
3. Структура организации.
4. Иерархическая структура работ.
5. Метрики сложности ПО.
6. Сетевая диаграмма.
7. PERT- диаграмма.
8. Диаграмма Ганта.
9. Задачи управления проектами.
10. Структура организации-исполнителя проекта.
11. Объектно-реляционный преобразователь.

Раздел 2. Образцы проектирования. Принципы создания удобного пользовательского интерфейса. Разработка различных уровней Web-приложений в J2EE

1. Аспектно-ориентированной программирование.

2. Web-службы.
3. Развитие технологий J2EE.
4. Общая схема архитектуры Web-приложений на основе Struts.
5. Класс действия.
6. Форма действия.
7. Управление приложениями.
8. Архитектура, основанная на службах.
9. Процесс стандартных описаний в форматах, основанных на XML.
10. Общая архитектура Web-приложений.
11. Компонент данных.
12. Сеансовый компонент.
13. Протокол HTTP.
14. Уровень бизнес-логики и модели данных в J2EE.
15. Компоненты, управляемые сообщениями.
16. Первичный ключ.
17. Удаленный интерфейс.
18. Локальный интерфейс.
19. Удаленный исходный интерфейс.

6.2. Оценочные средства для проведения промежуточной аттестации (зачета)

6.2.1. Примерный перечень вопросов/заданий к зачету:

1. Понятие инженерии программного обеспечения. Методы инженерии ПО.
2. Технологический процесс создания ПО, модели процесса создания ПО.
3. Разработка предложений по созданию ПО.
4. Требования к программному обеспечению.
5. Модели программных систем.
6. Прототипирование программных систем.
7. Формальные спецификации программного обеспечения.
8. Архитектурное проектирование.
9. Проблемно-зависимые архитектуры. Архитектура распределённых систем.
10. Процесс объектно-ориентированного проектирования.
11. Проектирование систем реального времени.
12. Проектирование интерфейса пользователя.
13. Верификация программного обеспечения.
14. Аттестация программного обеспечения.
15. Статический анализ программ.
16. Тестирование программного обеспечения.
17. Аттестация критических систем.
18. Оценка стоимости программного продукта.
19. Наследуемые системы.
20. Приведите примеры рекурсивных объектов и явлений. Обоснуйте проявление рекурсивности.
21. Почему при правильной организации рекурсивные вызовы не зацикливаются?
22. Почему не отождествляются совпадающие идентификаторы при многократных рекурсивных вызовах?
23. Почему рекурсивные обращения завершаются в порядке, обратном вызовам этих обращений?
24. Чем ограничено при выполнении программы количество рекурсивных вызовов?

25. Какой из методов в программировании является более эффективным – рекурсивный или итерационный?
26. Почему указатель не может существовать как самостоятельный тип?
27. С какой целью в программе может быть использован указатель типа void?
28. Что будет являться результатом разыменования указателя типа void без приведения типов?
29. Как изменится значение указателя после применения к нему операции инкремента (декремента)?
30. Почему для указателей определены сложение и вычитание только с целыми константами?
31. В чем отличие указателя на константу от указателя-константы?
32. Два указателя разных типов указывают на одно и то же место в памяти. Сравните результаты операций разыменования и взятия адреса с такими указателями. Сравните значения указателей.
33. Если объект занимает в памяти несколько байтов, то какой адрес является значением указателя на этот объект?
34. Каким образом при разыменовании указателей становится известно, сколько байтов памяти доступно?
35. Почему запрещены арифметические операции над указателями на функции?
36. Почему в описании указателя на функцию необходимы круглые скобки при имени указателя?
37. Может ли функция возвращать значение типа указатель? Если да, то как объявляется прототип такой функции?
38. Могут ли параметрами функции быть указатели на объекты? Если да, то как происходит передача фактических параметров при вызове функции?
39. Могут ли параметрами функции быть указатели на функции? Если да, то как происходит передача фактических параметров при вызове функции?
40. Как понимается следующее объявление:
`float>(*func)(int(*pf)(char),float);?`
41. В чем отличие результатов вызова функции через указатель с последующим разыменованием указателя и без разыменования указателя?
42. С какой целью в программах используют указатели на указатели?
43. Что будет являться результатом однократной операции разыменования указателя, реализующего многочисленное перенаправление?
44. Для чего в программе необходима инициализация указателя перед первым его использованием?
45. С какой целью в прототипах функций с переменным числом параметров должны быть указаны обязательные параметры?
46. Как в функции с переменным числом параметров осуществляется доступ к списку неизвестных параметров?
47. Почему для доступа к списку неизвестных параметров достаточно знать адрес хотя бы одного обязательного параметра?
48. Почему ошибки, связанные с некорректным использованием указателей, относятся к наиболее трудноустраняемым?
49. Почему в C++ не выполняется операция прямого присваивания значения строке?
50. Почему символ и строка, состоящая из одного символа, занимают разный объем памяти?
51. Почему в функции `scanf("%s",string);` не указывается обращение к переменной по адресу?
52. Допустима ли операция сравнения над символами? Если да, то каким образом определены отношения "больше" и "меньше"?

53. Какая из функций, gets или puts, заносит в поток управляющий символ '\n' и с какой целью?
54. Можно ли выполнить присваивание символьной переменной числового значения? Почему?
55. В чем различия результатов вывода символьной переменной со спецификаторами %d и %c?
56. Что будет являться результатом работы функции побайтового копирования строк, если длина строки-источника превосходит допустимый размер строки-приемника?
57. Что будет являться результатом работы функции побайтового копирования строк, если длина строки-источника меньше размера строки-приемника?
58. Почему при сравнении строк важен регистр символов?
59. Как сравниваются строки разной длины?
60. Какие возможны последствия при обращении к неинициализированной строке?
61. Почему функция изменения регистра символов строки может некорректно работать с кириллицей?
62. Почему обращения к строке через ее имя и через указатель эквивалентны?
63. Почему в качестве параметра функции передается адрес строки, а не сама строка символов?
64. Возможно ли применение операций инкремента и декремента к указателю на строку? Если да, то что будет адресовать полученный указатель?
65. Почему при формировании строки без использования стандартных функций необходимо дописывать символ конца строки? Почему этого не требуется при считывании строк с клавиатуры?
66. Какие возможны ошибки в программе при некорректной работе со строками?
67. Для защиты строки от изменения объявляется указатель на константу или указатель-константа? Почему?
68. Почему в программе на C++ необходимо, чтобы был известен размер массива?
69. Можно ли выполнить прямое присваивание массивов объявленных так: `int x[10], y[10];`?
70. Когда, с какой целью и почему возможно объявление безразмерных массивов?
71. В чем отличие обращения к элементам массива с помощью индексированного имени и посредством арифметики с указателями?
72. Может ли значение элемента массива использоваться в качестве индекса другого элемента массива?
73. Эквивалентны ли для массива `mas` следующие обращения и почему: `mas` и `&mas[0]`?
74. Какие ограничения распространяются на тип массива?
75. Каким образом можно определить объем памяти, выделяемой под массив?
76. Каким образом можно составить выражение для генерации массива случайными целыми числами на заданном промежутке?
77. Какие классы задач предполагают изменение значений элементов массива?
78. Какие классы задач предполагают только изменение порядка следования элементов в массиве?
79. Каким образом можно выполнять обход массива?
80. Почему в алгоритме циклического сдвига элементов массива важен порядок смещения элементов?
81. Чем различаются алгоритмы поиска первого и последнего минимального (максимального) элемента в массиве?
82. Почему существует большое количество алгоритмов сортировок?
83. С какой целью используются простые сортировки, если они характеризуются малой эффективностью?

84. Чем отличается принцип сортировки по неубыванию (невозрастанию) от сортировки по возрастанию (убыванию)?
85. На каких наборах исходных данных проявляется эффективность алгоритмов простых сортировок по сравнению друг с другом?
86. В чем заключается улучшение метода шейкер-сортировки по сравнению с пузырьковой сортировкой?
87. Почему в программе на C++ при объявлении двумерного массива необходимо, чтобы был известен размер по каждому измерению массива?
88. Можно ли выполнить прямое присваивание двумерных массивов?
89. Когда, с какой целью и почему возможно объявление безразмерных массивов? С одним безразмерным измерением?
90. В чем отличие обращения к элементам двумерного массива с помощью индексированного имени и посредством арифметики с указателями?
91. Эквивалентны ли для массива `mas` следующие обращения и почему: `mas` и `&mas[0][0]`?
92. Возможно ли в двумерном массиве `mas` обращение к элементу `&mas[0]`? Почему?
93. Приведите возможные обращения к элементу двумерного массива, аналогичные обращению `mas[i][j]`.
94. Какие ограничения распространяются на тип массива?
95. Каким образом можно определить объем памяти, выделяемой под двумерный массив?
96. В чем принципиальное отличие задач сортировок двумерных и одномерных массивов?
97. Каким образом оформляется прототип функции, чтобы изменения, выполненные с элементами массива, были сохранены после завершения работы функции?
98. Приведите возможные обращения к элементу трехмерного массива, аналогичные обращению `mas[i][j][k]`.
99. В чем причина неудобства использования массивов слишком больших измерений в программах?
100. При решении каких прикладных задач используются многомерные массивы? Отдельно приведите примеры для массивов с измерением два и более.
101. В чем принципиальное отличие типов массив и структура?
102. Как располагаются в памяти элементы структуры?
103. Почему размер структуры не всегда совпадает с суммарным размером ее полей?
104. Для моделирования каких данных целесообразно использовать структуры?
105. Какими способами можно обратиться к данным структуры?
106. В чем отличие прямого и косвенного доступа к полям структуры?
107. Всегда ли возможно выполнить напрямую операцию присваивания значений объектов структуры с одинаковым набором полей?
108. При каком объявлении структурных объектов возможно выполнить напрямую операцию присваивания значений объектов структуры?
109. Для моделирования каких данных целесообразно использовать массив структур?
110. Какие данные о структуре содержит указатель на эту структуру?
111. Какие ограничения накладываются на тип элемента структуры?
112. Возможно ли в качестве типа элемента структуры использовать указатель на другую структуру?
113. Как выполняется инициализация указателя на структуру?
114. Как выполняются операции инкремента и декремента над указателями на структуры?
115. Какими способами можно обратиться к данным структуры, используя указатели?
116. Каким образом необходимо передать структуру в качестве параметра функции, чтобы сохранить изменения, совершаемые функцией с данной структурой?

117. Что возвращается в качестве значения функции, тип которой объявлен как структура?
118. Что возвращается в качестве значения функции, тип которой объявлен как указатель на структуру?
119. В чем отличия текстовых и двоичных файлов с точки зрения представления данных?
120. Почему поток ввода-вывода не зависит от конкретного устройства?
121. Для чего необходима буферизация при потоковом и форматированном вводе-выводе?
122. С какой целью предусмотрены режимы открытия файлов и почему их такое многообразие?
123. Каковы могут быть причины ошибок при открытии файлов?
124. Какие значения возвращает функция открытия файла в зависимости от результата?
125. Каким образом в программе происходит проверка достижения конца файла?
126. Может ли один и тот же указатель на файл одновременно связан с несколькими файлами? Почему?
127. Может ли один и тот же файл одновременно быть открыт для чтения и для записи?
128. Можно ли один и тот же файл открыть несколько раз, не закрывая после каждого открытия?
129. Сохранится ли информация в файле, если его не закрыть в программе с помощью функции?
130. В чем основные отличия в организации символьного, строкового, блокового и форматированного ввода-вывода в файлы?
131. Почему ввод-вывод в файлы в потоковом режиме аналогичен работе с другими устройствами ввода-вывода?
132. С какой целью предусмотрены режимы открытия файлов и почему их такое многообразие?
133. Каковы могут быть причины ошибок при открытии файлов в потоковом режиме?
134. Какие значения возвращает функция открытия файла в потоковом режиме в зависимости от результата?
135. Каким образом в программе происходит проверка достижения конца файла?
136. Может ли один и тот же файл одновременно быть открыт для чтения и для записи?
137. Можно ли один и тот же файл открыть несколько раз, не закрывая после каждого открытия?
138. Сохранится ли информация в файле, если его не закрыть в программе с помощью функции? Обоснуйте ответ.
139. В чем принципиальное отличие ввода-вывода в файлы при потоковом и стандартном режимах?
140. Для чего используется динамическая память в программировании?
141. Какая область памяти выделяется под размещение динамических данных?
142. Как долго хранятся данные в динамической памяти?
143. Какие возможны варианты доступа к динамической памяти?
144. Что возвращает операция выделения динамической памяти в случае успешного выполнения?
145. Что возвращает операция выделения динамической памяти, если участок требуемого размера не может быть выделен?
146. Почему тип функций выделения динамической памяти определен как `*void`?
147. Почему при завершении работы с динамической памятью ее необходимо освободить? Какие могут быть последствия для работы программы, если не освободить динамическую память?
148. Существуют ли ограничения на данные при применении к ним операции или функции освобождения динамической памяти?

6.2.2. Примерные тестовые задания к зачету

Вариант 1

1.	Какую технологию программирования следует использовать при разработке систем интерфейсного приложения	<ul style="list-style-type: none"> 1) модульное программирование 2) структурное программирование 3) процедурное программирование 4) объектно-ориентированное программирование
2.	Событийным называется такое программирование, при котором . . .	<ul style="list-style-type: none"> 1) в программе описываются события реального мира 2) обрабатываются временные отрезки 3) программы реагирует на действие пользователя за пультом 4) программа не реагирует на ошибки данных
3.	Понятие «декомпозиция задачи» подразумевает:	<ul style="list-style-type: none"> 1) запись решающего алгоритма с помощью подпрограмм 2) логический анализ задачи с целью её представления как совокупности связанных более простых подзадач 3) представление задачи в виде блок-схемы алгоритма 4) запись решающего алгоритма на модульном алгоритмическом языке
4.	Объектно-ориентированным программированием называется такой способ разработки программ, при котором:	<ul style="list-style-type: none"> 1) программа построена из модулей, являющихся объединением данных и методов их обработки 2) программа ориентирована на обработку массивов 3) в процессе компиляции создаются объектные модули типа .obj 4) программа построена из модулей, объективно отражающих решающий алгоритм
5.	Событийным называется такое программирование, при котором . . .	<ul style="list-style-type: none"> 1) в программе описываются события реального мира 2) обрабатываются временные отрезки 3) программа не реагирует на ошибки данных 4) программы реагирует на действие пользователя за пультом
6.	Под «свойством» объекта подразумевают . . .	<ul style="list-style-type: none"> 1) значения, являющиеся параметрами объекта 2) способность объекта принимать данные от внешних источников информации 3) наличие процедур обработки данных 4) возможность реагирования на сигналы прерывания

7.	Понятие « события объекта» означает_	<ul style="list-style-type: none"> 1) нажатий клавиши пользователем 2) перемещение мыши пользователем 3) предусмотренное разработчиком объекта прерывание в операционной среде, на которое способен реагировать объект. 4) выключение компьютера
8.	Метод интерпретации имеет следующие преимущества над методом компиляции:	<ul style="list-style-type: none"> 1) строит модули в машинных кодах 2) более высокое быстродействие 3) позволяет выполнять программу из Интернета, независимо от типа процессора 4) обеспечивает наиболее простой и надёжный поиск ошибок
9.	Объектный модуль:	<ul style="list-style-type: none"> 1) является выражением принципа объектно-ориентированного программирования 2) образуется автоматически как промежуточный результат в процессе компиляции 3) необходим для реализации принципа модульного программирования 4) служит исходными данными для компилятора
10.	Подпрограмма функция ...	<ul style="list-style-type: none"> 1) возвращает в точку вызова более одного значения 2) не возвращает в точку ни одного значения 3) возвращает в точку вызова только одно значение 4) любая подпрограмма, хранящаяся в системной библиотеке
11.	Компилятором называется машинная программа, которая...	<ul style="list-style-type: none"> 1) преобразует каждое предложение входного языка в машинные коды, выполняет их и затем переходит к следующему предложению, не сохраняя машинных кодов 2) преобразует все предложения входного языка в машинные коды при условии, что языком является ассемблер 3) преобразует все предложения входного языка в машинные коды и записывает результат на магнитный диск в виде загрузочного модуля 4) преобразует входной язык в символические коды на языке MSIL

12.	Интерпретатором является машинная программа, которая...	<p>1) преобразует все предложения входного языка в машинные коды и записывает результат на магнитный диск в виде загрузочного модуля</p> <p>2) преобразует все предложения входного языка в машинные коды и передаёт их в сеть для использования другими программистами</p> <p>3) преобразует входной язык в символические коды на языке MSIL</p> <p>4) преобразует каждое предложение входного языка в машинные коды, выполняет их и затем переходит к следующему предложению, не сохраняя машинных кодов</p>
13.	Подпрограмма процедура ...	<p>1) возвращает только одно значение</p> <p>2) возвращает в точку вызова любое количество значений, в том числе ни одного</p> <p>2) возвращает в точку вызова только одно числовое значение</p> <p>3) это такая подпрограмма, вызов которой имеет форму оператора присваивания</p>
14.	Объектной называется программа ...	<p>1) записанная в машинном коде, эквивалентная исходному алгоритму, с подсоединёнными подпрограммами и полностью готовая к выполнению</p> <p>2) исходный текст которой записан с соблюдением принципов объектно-ориентированного программирования</p> <p>3) записанная в машинном коде, эквивалентная исходному алгоритму, но не готовая к выполнению, так как в ней отсутствуют библиотечные подпрограммы</p> <p>4) использующая классы и объекты языка C#</p>
15.	Преимущество метода компиляции над интерпретацией состоит в том, что ...	<p>1) машинная программа получается максимально короткой</p> <p>2) в программном тексте легче отыскивать грамматические ошибки</p> <p>3) программу можно передавать по сети любому компьютеру</p> <p>4) машинная программа получается максимально быстрой</p>

16.	Переносимость, как показатель качества программы, означает, что . . .	<ul style="list-style-type: none"> 1) программу разрешается копировать на внешние носители информации 2) программу можно перенести на другой компьютер без разрешения системного администратора 3) программа переносит попытки ввода некорректных данных 4) программа способна работать на различных аппаратных платформах или под управлением различных операционных систем
17.	Сопровождение программного продукта означает . . .	<ul style="list-style-type: none"> 1) обязательное присутствие разработчика во время эксплуатации 2) гарантированное исправление всех несоответствия программы функциям, зафиксированным в техническом задании 3) исправление всех недостатков, обнаруженных при эксплуатации 4) обязанность разработчика дополнять программу любыми функциями по желанию заказчика
18.	Термин CASE (Computer-Aided System Engineering) означает . . .	<ul style="list-style-type: none"> 1) технологический процесс проектирования, разработки и сопровождения программных систем с помощью компьютерных средств автоматизации. 2) использование сложных разветвлений в программе на базе case-инструкций 3) обработку аварийных ситуаций в case-ветвях алгоритма 4) использование программного контейнера типа «кейс»
19.	Объектно-ориентированное программирование означает . . .	<ul style="list-style-type: none"> 1) ориентации программы на движущиеся объекты 2) технологию программирования, направленную на использование программных конструкций, объединяющих в единое целое данные и методы их обработки 3) технологии типа 3D 4) использование объектных баз данных
20.	Укажите, какую технологию программирования следует использовать, чтобы программа смогла бы реагировать на внешние сигналы	<ul style="list-style-type: none"> 1) CASE- технологию 2) структурное программирование 3) объектно-ориентированное программирование 4) процедурное программирование

Вариант 2

1.	Понятие «методы объекта» означает	1) методическое описание свойств 2) приёмы описания объектов в программе 3) допустимые действия над объектом в целом 4) способы доступа к объекту
2.	Какие действия производит конструктор объекта ?	1) конструирует объект на этапе компиляции 2) резервирует область из раздела динамической памяти и заполняет начальными значениями свойств 3) строит экземпляр класса на этапе компиляции и делает его доступным для использования 4) описывает структуру объекта
3.	Создание объекта производится . . .	1) на этапе выполнения программы диспетчером памяти операционной системы при входе в блок, содержащий команду new 2) на этапе загрузки программы, представленной на языке MSIL (Microsoft Intermediate Language), средой CLR (Common Language Runtime) 3) на этапе обработки программы интерпретатором JIT (Just In Time) 4) на этапе компиляции системой VisualStudio, когда в тексте встречается конструктор объекта
4.	Блок «конструктор объекта» используется, чтобы . . .	1) зарезервировать память под объект при выполнении программы 2) установить начальные значения свойств объекта по умолчанию 3) проверить корректность методов объекта 4) освободить память, занимаемую объектом
5.	Понятие «полиморфизм» означает, что . . .	1) объект может иметь различные формы представления на экране 2) объекты одного класса могут иметь различные свойства 3) объекты одного класса могут иметь различные конструкторы 4) объекты одного класса могут обладать различными алгоритмами, хотя имена методов совпадают

6.	Понятие «перегрузка методов» означает . . .	<p>1) наличие в одном классе двух и более процедур, имеющих одно и то же имя, но различающихся количеством или типам формальных параметров</p> <p>2) слишком большое количество формальных параметров, приводящее к аварийному останову программы</p> <p>3) получение в процессе работы слишком большого числа, выходящего за пределы разрядной сетки</p> <p>4) возникновение ситуации, когда при вызове метода количество фактических параметров превышает число формальных</p>
7.	Понятие «платформа .NET Framework» означает . . .	<p>1) систему программирования на языке C#</p> <p>2) библиотеку системных подпрограмм, обеспечивающих совместимость компьютеров, подключённых к сети.</p> <p>3) электронную плату, обеспечивающую работу в сети</p> <p>4) электронные блоки для подключения к Интернету</p>
8.	Что представлено в каталоге ? pFiles.bpr pFiles.cpp pFiles.exe uFile.cpp uFile.dfm uFile.h	<p>1) приложение на C#</p> <p>2) приложение на C++</p> <p>3) приложение для Delphi</p> <p>4) приложение Visual Basic</p>
9.	Что представлено в каталоге ? dbStudent.sdf dbStudentDataSet.Designer.cs dbStudentDataSet.xsc frmOneTbl.cs frmOneTbl.Designer.cs OneTbl.csproj OneTbl.csproj.user Program.cs	<p>1) приложение на C#</p> <p>2) приложение на C++</p> <p>3) приложение для Delphi</p> <p>4) приложение Visual Basic</p>
10.	Переменная типа DialogResult используется для анализа результата, возвращаемого диалоговым окном и принимает тип..	<p>1) real</p> <p>2) sntring</p> <p>3) MessageBoxButtons</p> <p>4) MessageBoxIcon</p>
11.	Фиксированный формат числа имеет такие преимущества над плавающим форматом . . .	<p>1) больший динамический диапазон представления чисел</p> <p>2) более эффективное распределение памяти</p> <p>3) большая скорость выполнения арифметических операций</p> <p>4) повышенная точность вычислений</p>

12.	По инструкции объявления типа данных double X компьютер производит действия . . .	<ul style="list-style-type: none"> 1) резервирует в основной памяти 8 байтов и записывает в них начальное значение переменной X 2) резервирует в основной памяти 8 байтов, но ничего в них не записывает 3) резервирует в 8 байтов в расширенной памяти 4) отводит 4 байта в основной памяти
13.	Переменная является статической, если . . .	<ul style="list-style-type: none"> 1) под неё отводится основная память на этапе компиляции 2) под неё отводится расширенная память на этапе компиляции 3) под неё отводится расширенная память на этапе выполнения 4) её значение не изменяется при выполнении программы
14.	Переменная является динамической, если . . .	<ul style="list-style-type: none"> 1) её значение изменяется при выполнении программы 2) под неё отводится расширенная память на этапе компиляции 3) под неё отводится основная память на этапе компиляции 4) под неё отводится расширенная память на этапе выполнения
15.	Укажите значение параметра <code>keyPressEventArgs</code> е обработчика события <code>KeyPress</code> , которое используется для блокировки клавиатуры	<ul style="list-style-type: none"> 1)ToString 2)KeyChar 3)Handled 4)Equals
16.	Укажите значение параметра <code>keyPressEventArgs</code> е обработчика события <code>KeyPress</code> , которое используется для блокировки клавиатуры	<ul style="list-style-type: none"> 1)ToString 2)KeyChar 3)Handled 4)Equals
17.	Какие значения может принимать параметр <code>keyPressEventArgs</code> е ?	<ul style="list-style-type: none"> 1)char 2) byte 3)как char так и byte 4)string
18.	Укажите тип данных, не относящийся к пространству имён VisualStudio	<ul style="list-style-type: none"> 1) Int32 2) Float 3) Double 4) String
19.	Какой диапазон значений в пространстве имён VisualStudio определяет тип <code>Int16</code>	<ul style="list-style-type: none"> 1) -128 ...+127 2) -32768 ... 32767 3) -2 147 383 648 .. 2 147 383 647 4) -2 в степени 63 .+. 2 в степени 63
20.	Какой диапазон значений в пространстве имён VisualStudio определяет тип <code>Int32</code>	<ul style="list-style-type: none"> 1) -128 ...+127 2) -32768 ... 32767 3) -2 147 383 648 .. 2 147 383 647 4) -2 в степени 63 .+. 2 в степени 63

Вариант 3

1.	Какой диапазон значений в пространстве имён VisualStudio определяет тип Int64	<ul style="list-style-type: none"> 1) -128 ...+127 2) -32768 ... 32767 3) -2 147 383 648 .. 2 147 383 647 4) -2 в степени 63 .+. 2 в степени 63
2.	Какой диапазон значений в пространстве имён VisualStudio определяет тип Double	<ul style="list-style-type: none"> 1) -1.5E+45..3.4 E+38 2) -5.0E+324..1.7 E+308 3) -2 147 383 648 .. 2 147 383 647 4) -2 в степени 63 .+. 2 в степени 63
3.	Какой диапазон значений в пространстве имён VisualStudio определяет тип Single	<ul style="list-style-type: none"> 1) -1.5E+45..3.4 E+38 2) -5.0E+324..1.7 E+308 3) -2 147 383 648 .. 2 147 383 647 4) -2 в степени 63 .+. 2 в степени 63
4.	Какой диапазон значений в пространстве имён VisualStudio определяет тип String	<ul style="list-style-type: none"> 1) 32 байта 2) 16 батов 3) 255 символов 4) 65535 символов
5.	Какие данные в пространстве имён VisualStudio выражает тип Char	<ul style="list-style-type: none"> 1) 16 байта 2) 8 батов 3) 1 символ ANSI 4) 1 символ UNICOD
6.	Компонент ListBox используется для ...	<ul style="list-style-type: none"> 1) ввода с экрана значений элементов 2) ввода с экрана значений элементов коллекции 3) выбора одного предложения из предлагаемого списка 4) ввода с экрана списка объектов
7.	Компонент Label позволяет ...	<ul style="list-style-type: none"> 1) вводить числовые данные 2) вводить символьные данные 3) организовывать разветвления 4) выводить надпись на экран
8.	Компонент Panel –это ...	<ul style="list-style-type: none"> 1) графический контейнер для нескольких графических элементов интерфейса 2) программный контейнер, локализирующий несколько описанных в нём функций 3) программный контейнер, локализирующий несколько описанных в нём переменных 4) программная конструкция для описания объектов
9.	По оператору X=Y; компьютер ...	<ul style="list-style-type: none"> 1) записывает значение переменной Y в ячейку X 2) записывает символ Y в ячейку X 3) сравнивает переменные X и Y 4) приводит тип переменной Y к типу переменной X
10.	Оператор Console.Beep(); Обеспечивает	<ul style="list-style-type: none"> 1) вывод слова Беер на черный экран 2) вывод звукового сигнала 3) выдачу звука с частотой Беер 4) печать приветствия на главной форме проекта

11.	Как следует передать значение строковой переменной string S в переменную целого типа int N :	1) N=S; 2) N=Convert.ToInt32(S); 3) N=Convert.ToInt(s); 4) Move (S) To (N)
12.	Компонентом «Главное меню» пользуются, чтобы;	1) разместить на форме большое количество пунктов, максимально сэкономив визуальное пространство 2) получить наиболее быстрый доступ пунктам 3) упростить программирование 4) получить наиболее эффективную программу
13.	Контекстным меню называется такое меню ...	1) смысл и содержание которого зависит от контекста справки 2) содержание которого зависит от окружающих визуальных компонентов 3) которое вызывается двойным щелчком по компоненту 4) которой управляет контекстом WEB-сайта
14.	Управление сценарием диалога при помощи меню имеет такие преимущества по отношению к кнопочной форме:	1) более быстрый доступ к пунктам 2) более понятный интерфейс 3) позволяет разместить практически неограниченный сценарий диалога 4)упрощение программирования
15.	Выпадающее меню предпочтительнее использовать по отношению к обычному меню, потому что...	1) оно тесно связано с конкретным элементом интерфейса и поэтому его функции более понятны 2) оно запускается правой клавишей 3) работает быстрее 4) может быть больше по объёму
16.	Доступ к пункту главного меню можно получить, рассматривая пункт как ...	1) элемент коллекции 2) как элемент массива 3) как элемент оператора цикла типа for 4) как элемент оператора цикла типа foreach
17.	Какой из перечисленных методов не относится к классу ArrayList	1) Add 2)Clear 3)Remove 4) Substract
18.	Какой метод класса ArrayList добавляет элементов к концу коллекции	1) Add 2)Clear 3)Remove 4)Substract
19.	Какой метод класса ArrayList вставляет элемент в его коллекцию	1) Add 2)Clear 3)Remove 4) Insert
20.	Какой метод класса ArrayList вставляет в его коллекцию фрагмент другой коллекции	1) Insert 2)Clear 3)Remove 4) InsertRange

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

6.3.1. Критерии оценок промежуточной аттестации (зачета)

Примерная шкала оценивания знаний по вопросам/выполнению заданий зачета:

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 / курсовой проект 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. Емельянов, С. Г. Автоматизированные нечетко-логические системы управления : монография / С.Г. Емельянов, В.С. Титов, М.В. Бобырь. — Москва: ИНФРА-М, 2021. — 175 с. — (Научная мысль). - ISBN 978-5-16-009759-6. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1167848> (дата обращения: 02.09.2021). – Режим доступа: по подписке.
2. Поляков, А. Е. Электрические машины, электропривод и системы интеллектуального управления электротехническими комплексами: учебное пособие / А.Е. Поляков, А.В. Чесноков, Е.М. Филимонова. — Москва: ФОРУМ: ИНФРА-М, 2021. — 224 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-720-6. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1209815> (дата обращения: 02.09.2021). – Режим доступа: по подписке.
3. Москаленко, В. В. Системы автоматизированного управления электропривода: учебник / В.В. Москаленко. — Москва: ИНФРА-М, 2021. — 208 с. — (Среднее профессиональное образование). - ISBN 978-5-16-005116-1. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1157271> (дата обращения: 02.09.2021). – Режим доступа: по подписке.

7.1.2. Дополнительная литература

1. Системные методы анализа и синтеза интеллектуально-адаптивного управления: монография / С. О. Крамаров, Ю. А. Смирнов, С. В. Соколов, В. Н. Таран. — Москва: РИОР: ИНФРА-М, 2021. — 238 с. — (Научная мысль). - ISBN 978-5-369-01571-1. - Текст:

электронный. - URL: <https://znanium.com/catalog/product/1243846> (дата обращения: 02.09.2021). – Режим доступа: по подписке.

7.1.3. Учебно-методическое обеспечение

1. Учебно-методические материалы для проведения самостоятельной работы по учебной дисциплине «Прикладное программирование»: учебное пособие/ Ю.В. Ильюшин. – СПб. – 2018. – 8 с.
2. Методические рекомендации для выполнения курсовой работы (проекта) по учебной дисциплине «Прикладное программирование»: учебное пособие/ Ю.В. Ильюшин. – СПб. – 2018. – 6 с.
3. Учебно-методические разработки для проведения практических занятий по учебной дисциплине «Прикладное программирование»: учебное пособие/ Ю.В. Ильюшин. – СПб. – 2018. – 118 с.

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

1. Научная Электронная Библиотека <http://www.e-library.ru>.
2. ЭБС «Лань» <http://e.lanbook.com/>.
3. ЭБС «Znanium» <http://znanium.com/>.
4. ЭБС «IPRbooks» <http://www.iprbookshop.ru>.
5. Портал «Гуманитарное образование» <http://www.humanities.edu.ru/>.
6. Федеральный портал «Российское образование» <http://www.edu.ru/>.
7. Научная электронная библиотека «Scopus» <https://www.scopus.com/>
8. Поисковые системы Yandex, Rambler, Yahoo и др.
9. Электронная библиотека Российской Государственной Библиотеки (РГБ).
10. Электронная библиотека учебников: <http://studentam.net/>.
11. Электронно-библиотечная система <http://www.sciteclibrary.ru/>.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Материально-техническое оснащение аудиторий

1. Аудитория для проведения лекционных занятий и практических 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 фломастер – 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 занятий и практических 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)

4. MathCad Education, Договор №1134-11/12 от 28.11.2012 "На поставку программного обеспечения"

5. LabView Professional, ГК №1142912/09 от 04.12.2009 "На поставку программного обеспечения"