

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"Российский государственный гуманитарный университет"
(РГГУ)

ИНСТИТУТ ИНФОРМАЦИОННЫХ НАУК И ТЕХНОЛОГИЙ БЕЗОПАСНОСТИ
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И БЕЗОПАСНОСТИ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

По направлению подготовки 10.03.01 «Информационная безопасность»
профиль: Организация и технология защиты информации
Уровень квалификации выпускника *бакалавр*

Форма обучения *очная*

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2017

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Рабочая программа дисциплины

Составитель:

Старший преподаватель

кафедры информационных технологий и систем,

Е.П. Охапкина

Ответственный редактор

к.т.н., доцент А.А. Роганов

УТВЕРЖДЕНО

Протокол заседания

кафедры информационных технологий и систем

№ 3 от 24.01.2017г.

ОГЛАВЛЕНИЕ

1. Пояснительная записка

1.1 Цель и задачи дисциплины (*модуля*)

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения по дисциплине (*модулю*)

1.3. Место дисциплины в структуре образовательной программы

2. Структура дисциплины (*модуля*)

3. Содержание дисциплины (*модуля*)

4. Образовательные технологии

5. Оценка планируемых результатов обучения

5.1. Система оценивания

5.2. Критерии выставления оценок

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине (*модулю*)

6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Список источников и литературы

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

7. Материально-техническое обеспечение дисциплины (*модуля*)

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья

9. Методические материалы

9.1. Планы практических (семинарских, лабораторных) занятий

9.2. Методические рекомендации по подготовке письменных работ

9.3. Иные материалы

Приложения

Приложение 1. Аннотация дисциплины

Приложение 2. Лист изменений

| | | |
|--|--|--|
| | | <p>для построения высоко эффективных приложений;</p> <p>Владеть: базовыми принципами программирования на языке высокого уровня; методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение программы и используемой памяти компьютера; основными управляющими структурами языков Python; методами сортировки и хранения данных; методами и подходами к динамическому выделению памяти в пространстве памяти компьютера.</p> |
|--|--|--|

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Языки программирования» относится к базовой части блока учебного плана по направлению подготовки «Информационная безопасность», профиль «Организация и технология защиты информации». Имеет индекс Б1.Б.20 в структуре учебного плана. Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения следующих дисциплин: математический анализ, линейная алгебра, информатика, технология и методы программирования.

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: «Информационные технологии», «Информационные технологии. Операционные системы», «Информационные процессы и системы. Вычислительные сети», «Безопасность операционных систем», «Информационные технологии. Автоматизированные системы», «Безопасность программного обеспечения», «Базы данных, системы управления базами данных».

2. Структура дисциплины (модуля)

Структура дисциплины (модуля) для очной формы обучения

Общая трудоёмкость дисциплины составляет 4 з.е., 144 ч., в том числе контактная работа обучающихся с преподавателем 56 ч., самостоятельная работа обучающихся 70 ч., промежуточная аттестация – 18 ч.

| № п/п | Раздел дисциплины/темы | Семестр | Виды учебной работы (в часах) | | | | | Самостоятельная работа | Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам) |
|-------|---|---------|-------------------------------|---------|----------------------|----------------------|--------------------------|------------------------|---|
| | | | контактная | | | | | | |
| | | | Лекции | Семинар | Практические занятия | Лабораторные занятия | Промежуточная аттестация | | |
| 1 | Раздел 1. Программирование реляционной базы данных на языке C++ | 3 | 4 | | | 6 | | 10 | Защита отчета по лабораторной работе №1 |
| 2 | Раздел 2. Основы программирования | 3 | 4 | | | 8 | | 10 | Защита отчета по |

| | | | | | | | | | |
|---|--|---|-----------|--|--|-----------|--|-----------|--|
| | на языке Python. | | | | | | | | лабораторной работе №2 |
| 3 | Раздел 3. Алгоритмы анализа данных с использованием библиотеки Scikit-Learn. | 3 | 4 | | | 8 | | 10 | Защита отчета по лабораторной работе №3 |
| 4 | Раздел 4. Программируемая библиотека TensorFlow. | 3 | 4 | | | 8 | | 20 | Защита отчета по лабораторной работе №4 |
| 5 | Раздел 5. Нейросетевое моделирование на языке Python. | 3 | 4 | | | 6 | | 20 | Защита отчета; по лабораторной работе №5 |
| | <i>Экзамен</i> | 3 | | | | | | 18 | Экзамен по билетам |
| | итого: | | 20 | | | 36 | | 18 | 70 |

3. Содержание дисциплины (модуля)

| № | Наименование раздела дисциплины | Содержание |
|---|--|---|
| 1 | Программирование реляционной базы данных на языке C++ | Понятие о реляционной базе данных. Специализированные подходы к функциональному программированию. |
| 2 | Основы программирования на языке Python. | Типы данных. Инициализация переменных. Логические выражения и инструкция if . Множественное ветвление. Циклические операторы. Работа со строковыми данными. Понятие о «словаре». Введение в функциональное программирование. |
| 3 | Алгоритмы анализа данных с использованием библиотеки Scikit-Learn. | Классификация. MNIST. Линейная регрессия. Градиентный спуск. Деревья принятия решений. Ансамблевое обучение и случайные леса. |
| 4 | Программируемая библиотека TensorFlow. | Работа с TensorFlow. Создание и управление графом. Реализация градиентного спуска. Визуализация графа и кривых обучения с использованием TensorBoard. |
| 5 | Нейросетевое моделирование на языке Python. | Введение в искусственные нейронные сети. |

4. Образовательные технологии

Образовательные технологии

| № п/п | Наименование раздела | Виды учебных занятий | Образовательные технологии |
|-------|--|---|---|
| 1 | 2 | 3 | 4 |
| 1. | Раздел 1. Программирование реляционной базы данных на языке C++ | Лекция №1. Понятие о реляционной базе данных. Практическое занятие №1. Самостоятельная работа Лекция №2. Специализированные подходы к функциональному программированию. Самостоятельная работа | Вводная лекция с использованием видеоматериалов Развернутая беседа с обсуждением доклада Консультирование по вопросам лекции №1 и №2 |
| 2. | Раздел 2. Основы программирования на языке Python. | Лекция №3. Типы данных. Инициализация переменных. Логические выражения и инструкция if . Множественное ветвление. Практическое занятие №2. Самостоятельная работа | Лекция с применением методики интерактивного программирования Развернутая беседа с обсуждением доклада Консультирование по вопросам лекции №3 и |

| | | | |
|----|---|--|--|
| | | <p>Лекция №4. Циклические операторы. Работа со строковыми данными. Понятие о «словаре». Введение в функциональное программирование.</p> <p>Самостоятельная работа</p> | №4 |
| 3. | <p>Раздел 3. Алгоритмы анализа данных с использованием библиотеки Scikit-Learn.</p> | <p>Лекция №5. Классификация. MNIST. Линейная регрессия. Градиентный спуск.</p> <p>Практическое занятие №3.</p> <p>Самостоятельная работа</p> <p>Лекция №6. Деревья принятия решений. Ансамблевое обучение и случайные леса. лабораторное занятие №3.</p> <p>Самостоятельная работа</p> | <p>Лекция с применением методики интерактивного программирования</p> <p>Развернутая беседа с обсуждением доклада</p> <p>Консультирование по вопросам лекции №5 и №6</p> |
| 4. | <p>Раздел 4. Программируемая библиотека TensorFlow.</p> | <p>Лекция №7. Работа с TensorFlow. Создание и управление графом. Реализация градиентного спуска.</p> <p>Практическое занятие №4.</p> <p>Самостоятельная работа</p> <p>Лекция №8. Визуализация графа и кривых обучения с использованием TensorBoard.</p> <p>Самостоятельная работа</p> | <p>Лекция с применением методики интерактивного программирования</p> <p>Развернутая беседа с обсуждением доклада</p> <p>Консультирование по вопросам лекции №7 и №8</p> |
| 5. | <p>Раздел 5. Нейросетевое моделирование на языке Python.</p> | <p>Лекция №9. Введение в искусственные нейронные сети.</p> <p>Практическое занятие №5.</p> <p>Самостоятельная работа</p> <p>Лекция №10. Введение в искусственные нейронные сети.</p> <p>Самостоятельная работа</p> | <p>Лекция с применением методики интерактивного программирования</p> <p>Развернутая беседа с обсуждением доклада</p> <p>Консультирование по вопросам лекции №9 и №10</p> |

5. Оценка планируемых результатов обучения

5.1. Система оценивания

| Форма контроля | Макс. количество баллов | |
|---|-------------------------|-------------------|
| | За одну работу | Всего |
| Текущий контроль: - Лабораторная работа №1 | 12 баллов | 60 баллов |
| - Лабораторная работа №2 | 12 баллов | |
| - Лабораторная работа №3 | 12 баллов | |
| - Лабораторная работа №4 | 12 баллов | |
| - Лабораторная работа №5 | 12 баллов | |
| Промежуточная аттестация: зачет, экзамен | | 40 баллов |
| Итого за семестр (дисциплину) <i>Зачёт, экзамен</i> | | 100 баллов |

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

| 100-балльная шкала | Традиционная шкала | | Шкала ECTS |
|--------------------|---------------------|------------|------------|
| 95 – 100 | отлично | зачтено | A |
| 83 – 94 | | | B |
| 68 – 82 | хорошо | | C |
| 56 – 67 | удовлетворительно | | D |
| 50 – 55 | | | E |
| 20 – 49 | неудовлетворительно | не зачтено | FX |
| 0 – 19 | | | F |

5.2. Критерии выставления оценки по дисциплине

| Баллы/ Шкала ECTS | Оценка по дисциплине | Критерии оценки результатов обучения по дисциплине |
|-------------------------|---|---|
| 100-83/ A,B | «отлично»/ «зачтено (отлично)»/ «зачтено» | Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации. Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения. |

| Баллы/ Шкала ECTS | Оценка по дисциплине | Критерии оценки результатов обучения по дисциплине |
|-------------------------|--|---|
| | | <p>Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p> |
| 82-68/ С | «хорошо»/ «зачтено (хорошо)»/ «зачтено» | <p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей.</p> <p>Обучающийся правильно применяет теоретические положения при решении лабораторных задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами.</p> <p>Достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p> |
| 67-50/ D,E | «удовлетворительно»/ «зачтено (удовлетворительно)»/ «зачтено» | <p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает определённые затруднения в применении теоретических положений при решении лабораторных задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами.</p> <p>Демонстрирует достаточный уровень знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p> |
| 49-0/ F,FX | «неудовлетворительно»/ не зачтено | <p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении</p> |

| Баллы/ Шкала ECTS | Оценка по дисциплине | Критерии оценки результатов обучения по дисциплине |
|-------------------------|-------------------------|---|
| | | <p>лабораторных задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами. Демонстрирует фрагментарные знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p> |

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине (модулю)

Вопросы для подготовки к экзамену

1. Введение в Python. Переменные, константы. Типы данных. Операторы, выражения.
2. Ввод-вывод данных.
2. Библиотека Scikit-Learn.
3. Исключения в Python.
4. Ветвления и циклы в Python.
5. Строки. Базовые операции. Функции и методы строк.
6. Списки. Базовые операции. Функции и методы списков. Генераторы списков. Вложенные списки.
7. Кортежи. Базовые операции.
8. Множества. Базовые операции. Генераторы множеств.
9. Словари. Создание, базовые операции. Методы словарей.
10. Функции. Передача параметров в функцию. Области видимости переменных в Python.
11. Дополнительные возможности при работе с функциями.
12. MNIST.
13. Обучение двоичного классификатора на Python.
14. Показатели производительности алгоритма на Python.
15. Классификация на множестве классов с использованием Python.
16. Обучение моделей.
17. Оптимизация на Python: пакетный градиентный спуск, стохастический градиентный спуск.
18. Обучение и визуализация дерева принятия решений.
19. Алгоритм обучения CART. Коэффициент Джини. Понятие об энтропии.
20. Ансамблевое обучение. Бэггинг и вставка в Scikit-Learn.
21. Случайные леса на Python.
22. Framework TensorFlow. Создание графа.
23. Линейная регрессия с помощью TensorFlow.
24. Визуализация графа и кривых обучения с использованием TensorBoard.
25. Искусственные нейронные сети на Python.
26. Персептрон и его многослойная модификация на Python.
27. Обучение многослойного персептрона.

6. Учебно-методическое и информационное обеспечение дисциплины

- 6.1. Список источников и литературы

Литература

Основная

1. Трусов А. Н. Редактирование и внесение информации в XML-документы автоматизированных информационных систем [Текст] = Editing and entering information into XML-documents of automated information systems / А. Н. Трусов, П. Ю. Иванченко, Д. А. Кацуро // Программные продукты и системы. - 2017. - Т. 30, № 1. - С. 81-84. - Библиогр.: с. 84 (13 назв.). - ил.: 1 рис.

2. Белов А. В. (руководитель). Построение системы атрибутивного поиска в системах управления учетными или идентификационными доменами при внедрении интеграционных решений [Текст] / А. В. Белов, М. И. Нежурина, А. Д. Шестова // Научно-техническая информация. Сер. 2, Информационные процессы и системы. - 2017. - № 11. - С. 5-9. - Примеч. в сносках. - Библиогр.: с. 9 (5 назв.). - Ил.: 1 рис.

3. Колдаев Виктор Дмитриевич. Численные методы и программирование : Учебное пособие. - 1. - Москва; Москва: Издательский Дом "ФОРУМ" : ООО "Научно-издательский центр ИНФРА-М", 2017. - 336 с. - ВО - Бакалавриат. - ISBN 978-5-8199-0333-9.

4. Карпенков Степан Харланович. Современные средства информационных технологий: учебное пособие для студентов высших учебных заведений, обучающихся по направлениям подготовки дипломированных специалистов "Информатика и вычислительная техника" и "Информационные системы" / С. Х. Карпенков. - 2-е изд., испр. и доп. - Москва : КноРус, 2017. - 399, [1] с. : рис., табл. ; 22 см. - Библиогр.: с. 399-400. - ISBN 978-5-390-00393-0 : 318.50.

Дополнительная

1. Ступина Алена Александровна. Технология надежного программирования задач автоматизации управления в технических системах: ВО - Магистратура. - 1. - Красноярск: Сибирский федеральный университет, 2011. - 164 с. - ВО - Магистратура. - ISBN 9785763823547. Ссылка на ресурс: <http://new.znaniium.com/go.php?id=442655>

2. Веселовский А. М. Построение слабосвязанного программного обеспечения для оценивания качества проектных решений [Текст] / А. М. Веселовский // Программные продукты и системы. - 2011. - N 1. - С. 73-76. - Библиогр.: с. 76 (6 назв.). - ил.: 3 рис.

Перечень БД и ИСС

| №п/п | Наименование |
|------|---|
| | Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2017 г. Web of Science Scopus |
| | Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2017 г. Журналы Oxford University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis |
| | Компьютерные справочные правовые системы Консультант Плюс, Гарант |

7. Материально-техническое обеспечение дисциплины/модуля

Материально-техническая база включает учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Современный компьютерный класс оснащен

Перечень ПО

| №п /п | Наименование ПО | Производитель | Способ распространения (лицензионное или свободно распространяемое) |
|-------|-----------------------------|---------------|--|
| 1 | Microsoft Office 2010 | Microsoft | лицензионное |
| 2 | Windows XP | Microsoft | лицензионное |
| 3 | Kaspersky Endpoint Security | Kaspersky | лицензионное |

включающий наряду с компьютерами, подключёнными к сети Интернет, экран и проектор:

- Предустановленный дистрибутив Python;
- PyCharm или Jupyter Notebook.

Для проведения занятий лекционного типа предлагаются тематические иллюстрации в формате презентаций PowerPoint.

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

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

· для слепых и слабовидящих:

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

· для глухих и слабослышащих:

- лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;
- письменные задания выполняются на компьютере в письменной форме;
- экзамен проводится в письменной форме на компьютере; возможно проведение в форме тестирования.

· для лиц с нарушениями опорно-двигательного аппарата:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

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

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

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

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

- для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
 - в форме аудиофайла.
- для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;
 - в форме электронного документа;
 - в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих:
 - устройством для сканирования и чтения с камерой SARA CE;
 - дисплеем Брайля PAC Mate 20;
 - принтером Брайля EmBrailleViewPlus;
- для глухих и слабослышащих:
 - автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
 - акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - передвижными, регулируемые эргономическими партами СИ-1;
 - компьютерной техникой со специальным программным обеспечением.

9. Методические материал

9.1. Планы лабораторных занятий (*проверка сформированности компетенций ОПК-2, ОПК-7*):

Тема 1 (лабораторная работа 2 ч.). *Основы программирования на языке C.*

Задание:

1. Согласно заранее определенному варианту выполнить программирование операторов цикла и ветвления для решения алгебраической задачи;
2. Типизацию переменных задачи осуществить согласно принятым стандартам языков C/C++;
3. Подготовить развернутый отчет о проделанной работе.

Тема 2 (лабораторная работа 8 ч.). *Основы программирования на языке Python.*

Задание:

1. Согласно заранее определенному варианту выполнить объявление многомерного статического массива на языке Python;
2. Используя пользовательские функции, выполнить поиск и перестановку минимального и максимального элемента.

3. Подготовить развернутый отчет о проделанной работе.

Тема 3 (лабораторная работа 8 ч.). Алгоритмы анализа данных с использованием библиотеки Scikit-Learn.

Задание:

1. Согласно заранее определенному варианту выполнить программирование алгоритмов градиентного спуска, классификации и регрессионного моделирования с использованием прежде изученных операторов цикла и ветвления;
2. Подготовить развернутый отчет о проделанной работе.

Тема 4 (лабораторная работа 8 ч.). Программируемая библиотека TensorFlow.

Задание:

1. Согласно заранее определенному варианту выполнить построение графа решения с использованием Framework TensorFlow;
2. Осуществить визуализацию решения заданного варианта на базе TensorBoard.
3. Подготовить развернутый отчет о проделанной работе.

Тема 5 (лабораторная работа 8 ч.). Нейросетевое моделирование на языке Python.

Задание:

1. Согласно заранее определенному варианту построить многослойный персептрон;
2. Предусмотреть возможность обучения ИНС с/без учителя;
3. Подготовить развернутый отчет о проделанной работе.

9.2. Методические рекомендации по подготовке письменных работ

Отчет по проделанной работе должен быть изложен с соблюдением правил грамматики русского и английского языков (в случаях необходимости). При этом отражаемые результаты работы должны быть информативными, тезисного порядка. В отчет входят следующие обязательные разделы:

1. Титульный лист с полным указанием ведомственной принадлежности, названия ВУЗа, института, факультета, кафедры. Кроме того, полное точное название лабораторной работы, Ф.И.О. студента подготовившего отчет о результатах проделанной работы и Ф.И.О., должность, название кафедры преподавателя осуществляющего проверку и оценивание полученных результатов.
2. Содержание.
3. Введение.
4. Цели и задачи лабораторной работы.
5. Методы и технологии, применяемые для решения поставленных задач оформленные в виде отдельных этапов работы.
6. Выводы по работе.
7. Приложения.

Оформление отчета выполняется с использованием компьютерной верстки **LaTeX**. Отчет сохраняется и представляет для проверки в виде отдельного **pdf** файла. В имени файла указывается фамилия студента и номер выполненной работы.

АННОТАЦИЯ ДИСЦИПЛИНЫ

Дисциплина реализуется на факультете информационных систем и безопасности института информационных наук и технологий безопасности РГГУ, кафедрой информационных технологий и систем.

Цель дисциплины – приобретение знаний, навыков и умений в области высокоуровневых языков программирования, а также освоение современных алгоритмов анализа больших данных.

Задачи дисциплины:

1. Изучение базовых принципов программирования на языке высокого уровня Python;
2. Изучение специализированных технологий и методов программирования на языках C/C++ и Python для анализа и хранения данных;
3. Изучение главных управляющих структур языков Python;
4. Изучение функционального программирования на языке Python;
5. Приобретение навыков и умений по разработке алгоритмов в задачах анализа данных с использованием библиотеки Scikit-Learn;
6. Формирование навыков работы в программной библиотеке TensorFlow;
7. Приобретение навыков работы в среде PyCharm и Jupyter;
8. Сформировать представления о разработке эффективных приложений;
9. Дать представление о методике исследования больших данных;
10. Обучить основам построения самообучающихся алгоритмов на языке Python;

Знать: базовых принципов программирования на языке высокого уровня Python; главные управляющих структур языков Python; принципы объектно-ориентированного программирования; алгоритмы отладки разрабатываемых приложений; методы разработки эффективных приложений и оптимизации по времени и памяти; основы построения самообучающихся алгоритмов; специализированные методы обработки и хранения данных.

Уметь: применять специализированные методы обработки и хранения данных на языке Python; разрабатывать алгоритмы для решения задач последовательного и объектно-ориентированного программирования; применять принципы объектно-ориентированного программирования для построения высоко эффективных приложений;

Владеть: базовыми принципами программирования на языке высокого уровня; методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение программы и используемой памяти компьютера; основными управляющими структурами языков Python; методами сортировки и хранения данных; методами и подходами к динамическому выделению памяти в пространстве памяти компьютера.

По дисциплине (*модулю*) предусмотрена промежуточная аттестация в форме экзамена в *третьем семестре*.

Общая трудоемкость освоения дисциплины (*модуля*) составляет 4 зачетных единиц.

ЛИСТ ИЗМЕНЕНИЙ

| № | Текст актуализации или прилагаемый к РПД документ, содержащий изменения | Дата | № протокола |
|----|--|---------------|-------------|
| 1 | <i>Обновлен состав программного обеспечения (ПО), современных профессиональных баз данных (БД) и информационно-справочные систем (ИСС)</i> | 26.06.2017 г. | 4 |
| 2 | <i>Обновлена структура дисциплины (модуля) для очной формы обучения (2018 г.)</i> | 26.06.2018 г. | 12 |
| 3 | <i>Методическое обеспечение дисциплины</i> | 26.06.2018 г. | 12 |
| 4 | <i>Обновлена основная и дополнительная литература</i> | 26.06.2018 г. | 12 |
| 5 | <i>Обновлен состав программного обеспечения (ПО), современных профессиональных баз данных (БД) и информационно-справочные систем (ИСС)</i> | 26.06.2018 г. | 12 |
| 6 | <i>Обновлена структура дисциплины (модуля) для очной формы обучения (2019 г.)</i> | 26.06.2019 г. | 12 |
| 7 | <i>Обновлена основная и дополнительная литература</i> | 26.06.2019 г. | 12 |
| 8 | <i>Обновлен состав программного обеспечения (ПО), современных профессиональных баз данных (БД) и информационно-справочные систем (ИСС)</i> | 26.06.2019 г. | 12 |
| 9 | <i>Обновлена структура дисциплины (модуля) для очной формы обучения (2020 г.)</i> | 08.06.2020 | 12 |
| 10 | <i>Обновлена основная и дополнительная литература</i> | 08.06.2020 | 12 |
| 11 | <i>Обновлен раздел п.4 Образовательные технологии</i> | 08.06.2020 | 12 |
| 12 | <i>Обновлен состав программного обеспечения (ПО), современных профессиональных баз данных (БД) и информационно-справочные систем (ИСС)</i> | 08.06.2020 | 12 |

1. Состав программного обеспечения (ПО), современных профессиональных баз данных (БД) и информационно-справочных систем (ИСС) (2017 г.)

Перечень ПО

Таблица 1

| №п/п | Наименование ПО | Производитель | Способ распространения (лицензионное или свободно распространяемое) |
|------|---------------------------|------------------|--|
| 1 | MicrosoftOffice 2013 | Microsoft | лицензионное |
| 2 | Windows XP | Microsoft | лицензионное |
| 3 | KasperskyEndpointSecurity | Kaspersky | лицензионное |
| 4 | ОС «Альт Образование» 8 | ООО «Базальт СПО | лицензионное |

Перечень БД и ИСС

Таблица 2

| №п/п | Наименование |
|------|--|
| | Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2017 г. Web of Science Scopus |
| | Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2017 г. Журналы Oxford University Press |
| | Компьютерные справочные правовые системы Консультант Плюс, Гарант |

Составитель:

Старший преподаватель кафедры информационных технологий и систем, Е.П. Охупкина

2. Обновление структуры дисциплины (модуля) для очной формы обучения (2018 г.)**Структура дисциплины (модуля) для очной формы обучения**

Общая трудоёмкость дисциплины составляет 4 з.е., 144 ч., в том числе контактная работа обучающихся с преподавателем 56 ч., самостоятельная работа обучающихся 70 ч., промежуточная аттестация – 18 ч.

| № п/п | Раздел дисциплины/темы | Семестр | Виды учебной работы (в часах) | | | | | Самостоятельная работа | Формы текущего контроля успеваемости, форма промежуточной аттестации <i>(по семестрам)</i> |
|-------|--|---------|----------------------------------|---------|----------------------|----------------------|--------------------------|------------------------|--|
| | | | контактная | | | | | | |
| | | | Лекции | Семинар | Практические занятия | Лабораторные занятия | Промежуточная аттестация | | |
| 1 | Раздел 1. Программирование реляционной базы данных на языке С++ | 3 | 4 | | 6 | | | 10 | Защита отчета по лабораторной работе №1 |
| 2 | Раздел 2. Основы программирования на языке Python. | 3 | 4 | | 8 | | | 10 | Защита отчета по лабораторной работе №2 |
| 3 | Раздел 3. Алгоритмы анализа данных с использованием библиотеки Scikit-Learn. | 3 | 4 | | 8 | | | 10 | Защита отчета по лабораторной работе №3 |
| 4 | Раздел 4. Программируемая библиотека TensorFlow. | 3 | 4 | | 8 | | | 20 | Защита отчета по лабораторной работе №4 |
| 5 | Раздел 5. Нейросетевое моделирование на языке Python. | 3 | 4 | | 6 | | | 20 | Защита отчета; по лабораторной работе №5 |
| | <i>Экзамен</i> | 3 | | | | | 18 | | Экзамен по билетам |
| | итого: | | 20 | | 36 | | 18 | 70 | |

3. Обновление методического материала Планы лабораторных работ

| № | Название лабораторной работы | Число часов |
|----|--|-------------|
| 1 | Лабораторная работа №1. Особенности языка C++, не связанные с объектной ориентированностью | 2 |
| 2 | Лабораторная работа №2. Компиляция программ на языке C++ | 2 |
| 3 | Лабораторная работа №3. Введение в понятия класса и объекта | 2 |
| 4 | Лабораторная работа №4. Перегрузка функций | 2 |
| 5 | Лабораторная работа №5. Перегрузка операций | 4 |
| 6 | Лабораторная работа №6. Наследование | 2 |
| 7 | Лабораторная работа №7. Конструкторы и деструкторы | 4 |
| 8 | Лабораторная работа №8. Ключевые слова C++ | 4 |
| 9 | Лабораторная работа №9. Дружественные функции и классы | 4 |
| 10 | Лабораторная работа №10. Структуры и объединения | 4 |
| 11 | Лабораторная работа №11. Манипуляторы | 4 |
| 12 | Лабораторная работа №12. Двоичный ввод/вывод | 2 |
| | Итого | 36 |

Текущий контроль:

Темы рефератов:

- Объектно-ориентированные языки программирования. Их сходство и отличия
- Компиляторы объектно-ориентированных языков программирования. Их сходство и отличия
- Понятия классов и объектов
- Понятие перегрузки функций и операций
- Что такое наследование. Виды наследования
- Конструкторы и деструкторы. Их типы
- Ключевые слова языков программирования
- Дружественные функции и классы. Их отличие и способы использования
- Что такое манипуляторы. Их типы
- Отличие двоичного ввода/вывода от текстового
- Строки и контейнеры. Их отличия
- Указатели и функции. Как они связаны
- Что такое шаблоны в объектно-ориентированных языках
- Что такое потоки ввода/вывода
- Какие дополнительные средства языка C++ Вы знаете

Примерные вопросы для тестирования:

1. Что определяет класс? Чем обличается класс от объекта?
2. Можно ли объявлять массив объектов? А массив классов?
3. Разрешается ли объявлять указатель на объект? А указатель на класс?
4. Допускается ли передавать объекты в качестве параметров, и какими способами? А возвращать как результат?
5. Как называется использование объекта одного класса в качестве поля другого класса?
6. Является ли структура классом? Чем класс отличается от структуры?
7. Какие ключевые слова в C++ обозначают класс?
8. Объясните принцип инкапсуляции.
9. Что такое композиция?
10. Для чего используются ключевые слова public и private?
11. Можно ли использовать ключевые слова public и private в структуре?
12. Существуют ли ограничения на использование public и private в классе? А в структуре?
13. Обязательно ли делать поля класса приватными?

14. Что такое метод? Как вызывается метод?
15. Может ли метод быть приватный?
16. Как определить метод непосредственно внутри класса? А вне класса? Чем эти определения отличаются?
17. Можно в методах присваивать параметрам значения по умолчанию?
18. Что обозначается ключевым словом `this`?
19. Зачем нужны константные методы? Чем отличается определение константного метода от обычного?
20. Может ли константный метод вызываться для объектов-переменных? А обычный метод — для объектов-констант?
21. Объясните принцип полиморфизма.
22. Сколько места в памяти занимает объект класса? Как это узнать?
23. Каков размер «пустого» объекта?
24. Влияют ли методы на размер объекта?
25. Одинаков ли размер класса и аналогичной структуры?
26. Какие операции нельзя перегружать? Как вы думаете, почему?
27. Можно ли перегружать операции для встроенных типов данных?
28. Можно ли при перегрузке изменить приоритет операции?
29. Можно ли определить новую операцию?
30. Перечислите особенности перегрузки операций как методов класса. Чем отличается перегрузка внешним образом от перегрузки как метода класса?
31. Какой результат должны возвращать операции с присваиванием?
32. Как различаются перегруженная префиксная и постфиксная операции инкремента и декремента?
33. Что означает выражение `*this`? В каких случаях оно используется?
34. Какие операции не рекомендуется перегружать как методы класса? Почему?
35. Какие операции разрешается перегружать только как методы класса?
36. Дайте определение дружественной функции. Как объявляется дружественная функция? А как определяется?
37. Дайте определение конструктора. Каково назначение конструктора? Перечислите отличия конструктора от метода.
38. Сколько конструкторов может быть в классе? Допускается ли перегрузка конструкторов? Какие виды конструкторов создаются по умолчанию?
39. Может ли конструктор быть приватным? Какие последствия влечет за собой объявление конструктора приватным?
40. Приведите несколько случаев, когда конструктор вызывается неявно.
41. Как проинициализировать динамическую переменную?
42. Как объявить константу в классе? Можно ли объявить дробную константу?
43. Каким образом разрешается инициализировать константные поля в классе?
44. В каком порядке инициализируются поля в классе? Совпадает ли этот порядок с порядком перечисления инициализаторов в списке инициализации конструктора?
45. Какие конструкции C++ разрешается использовать в списке инициализации качестве инициализирующих выражений?
46. Какой вид конструктора фактически является конструктором преобразования типов?
47. Для чего нужны функции преобразования? Как объявить такую функцию в классе?
48. Как запретить неявное преобразование типа, выполняемое конструктором инициализации?
49. Какие проблемы могут возникнуть при определении функций преобразования?
50. Для чего служит ключевое слово `explicit`?
51. Влияет ли наличие целочисленных констант-полей на размер класса?
52. Разрешается ли объявлять массив в качестве поля класса. Как присвоить элементам массива начальные значения?

53. Сколько операндов имеет операция индексирования []? Какой вид результата должна возвращать эта операция?
54. Для чего нужны статические поля в классе? Как они определяются?
55. Как объявить в классе и проинициализировать статический константный массив?
56. Что такое выравнивание и от чего оно зависит? Влияет ли выравнивание на размер класса?
57. Дайте определение контейнера.
58. Какие виды встроенных контейнеров в C++ вы знаете?
59. Какие виды доступа к элементам контейнера вам известны?
60. Чем отличается прямой доступ от ассоциативного?
61. Перечислите операции, которые обычно реализуются для последовательного доступа к элементам контейнера.
62. Дайте определение итератора.
63. Можно ли реализовать последовательный доступ без итератора? В чем преимущества реализации последовательного доступа с помощью итератора?
64. Что играет роль итератора для массивов C++?
65. Что такое деструктор? Может ли деструктор иметь параметры?
66. Почему для классов-контейнеров деструктор надо писать явным образом?
67. Допускается ли перегрузка деструкторов?
68. Что такое «глубокое копирование» и когда в нем возникает необходимость?
69. Какое копирование осуществляет стандартный конструктор копирования?
70. Чем отличается копирование от присваивания?
71. Объясните, почему в операции присваивания требуется проверка присваивания самому себе?
72. Можно ли в качестве операции индексирования использовать операцию вызова функции ()? В чем ее преимущества перед операцией []?
73. Почему необходимо писать два определения операции индексирования? Чем они отличаются?
74. Дайте определение вложенного класса.
75. Можно ли класс-итератор реализовать как внешний класс? А как вложенный? В чем отличия этих методов реализации?
76. Может ли объемлющий класс иметь неограниченный доступ к элементам вложенного класса? А вложенный класс — к элементам объемлющего?
77. Ограничена ли глубина вложенности классов?
78. Можно ли определить вложенный класс внешним образом? Зачем это может понадобиться?
79. Каким образом вложенный класс может использовать методы объемлющего класса? А объемлющий — методы вложенного?
80. Что такое «запредельный» элемент, какую роль он играет в контейнерах?
81. Объясните, по каким причинам трудно написать универсальный контейнер, элементы которого могут иметь произвольный тип.
82. Назовите ключевые слова C++, которые используются для обработки исключений.
83. Исключение — это:
- 1) событие;
 - 2) ситуация;
 - 3) объект;
 - 4) ошибка в программе;
 - 5) прерывание;
84. Каким образом исключение генерируется?
85. Каковы функции контролируемого блока?
86. Что обозначается ключевым словом catch?
- 1) контролируемый блок;

- 2) блок обработки исключения;
- 3) секция-ловушка;
- 4) генератор исключения;
- 5) обработчик прерывания;
87. Какого типа может быть исключение?
88. Сколько параметров разрешается писать в заголовке секции-ловушки?
89. Какими способами разрешается передавать исключение в блок обработки?
90. Объясните, каким образом преодолеть ограничение на передачу единственного параметра в блок обработки.
91. Почему нельзя выполнять преобразования типов исключений при передаче в секцию-ловушку?
92. Напишите конструкцию, которая позволяет перехватить любое исключение.
93. Могут ли контролируемые блоки быть вложенными?
94. Зачем нужен «контролируемый блок-функция» и чем он отличается от обычного контролируемого блока?
95. Перечислите возможные способы выхода из блока обработки.
96. Каким образом исключение «передать дальше»?
97. Сколько секций-ловушек должно быть задано в контролируемом блоке?
98. Что такое «спецификация исключений»?
99. Что происходит, если функция нарушает спецификацию исключений?
100. Учитывается ли спецификация исключений при перегрузке функций?
101. Что такое «иерархия исключений»?
102. Существуют ли стандартные исключения? Назовите два-три типа стандартных исключений.
103. Поясните «взаимоотношение» исключений и деструкторов.
104. Объясните, зачем может понадобиться подмена стандартных функций завершения.
105. Какие виды нестандартных исключений вы знаете?
106. В чем отличие механизма структурной обработки исключений Windows от стандартного механизма?
107. Какие две роли выполняет наследование?
108. Какие виды наследования возможны в C++?
109. Чем отличается модификатор доступа `protected` от модификаторов `private` и `public`?
110. Чем открытое наследование отличается от закрытого и защищенного?
111. Какие функции не наследуются?
112. Сформулируйте правила написания конструкторов в производном классе.
113. Каков порядок вызова конструкторов? А деструкторов?
114. Можно ли в производном классе объявлять новые поля? А методы?
115. Если имя нового поля совпадает с именем унаследованного, то каким образом разрешить конфликт имен?
116. Что происходит, если имя метода-наследника совпадает с именем базового метода?
117. Сформулируйте принцип подстановки.
118. Когда выполняется понижающее приведение типов?
119. Объясните, что такое «срезка» или «расщепление».
120. Объясните, зачем нужны виртуальные функции.
121. Что такое связывание?
122. Чем «раннее» связывание отличается от «позднего»?
123. Какие два вида полиморфизма реализованы в C++?
124. Дайте определение полиморфного класса.
125. Может ли виртуальная функция быть дружественной функцией класса?
126. Наследуются ли виртуальные функции?
127. Каковы особенности вызова виртуальных функций в конструкторах и деструкторах?
128. Можно ли сделать виртуальной перегруженную операцию, например, сложение?

129. Может ли конструктор быть виртуальным? А деструктор?
130. Как виртуальные функции влияют на размер класса?
131. Как объявляется «чистая» виртуальная функция?
132. Дайте определение абстрактного класса.
133. Наследуются ли чистые виртуальные функции?
134. Можно ли объявить деструктор чисто виртуальным?
135. Чем отличается чистый виртуальный деструктор от чистой виртуальной функции?
136. Зачем требуется определение чистого виртуального деструктора?
137. Наследуется ли определение чистой виртуальной функции?
138. Приведите классификацию целей наследования.
139. Объясните разницу наследования интерфейса от наследования реализации.
140. Назовите причины, требующие разделения программ на части.
141. Дайте определение термина «единица трансляции»?
142. Чем отличается файл с исходным текстом от единицы трансляции?
143. Существуют ли в C++ конструкции, позволяющие идентифицировать отдельный модуль?
144. Какие способы сборки программы вы можете назвать?
145. Что такое «объектный модуль»? Программа, которая «собирает» объектные модули в программу, называется _____ ?
146. В чем заключается отличие аргумента «файл» от <файл> в директиве #include?
147. Что такое ODR?
148. Объясните, что такое «страж» включения и зачем он нужен.
149. Является ли интерфейс класса его определением?
150. Сколько определений класса может быть в единице трансляции?
151. Сколько определений класса может быть в многофайловой программе?
152. Чем отличаются стандартные заголовки <string>, <string.h> и <cstring>?
153. Объясните суть идиомы Pimpl.
154. Что такое делегирование и как его можно использовать для повышения степени инкапсуляции?
155. Каким образом глобальную переменную, определенную в одной единице трансляции, сделать доступной в другой единице трансляции? А константу?
156. Можно ли использовать слово extern при объявлении функций?
157. Как локализовать объявление функции в файле?
158. Чем отличается «внешнее» связывание от «внутреннего» связывания?
159. Что такое «спецификации компоновки»?
160. Какие объекты обладают внутренним связыванием по умолчанию?
161. Какие области видимости имен вы знаете?
162. Для чего используются пространства имен?
163. Чем отличаются именованные и неименованные пространства имен?
164. Могут ли пространства имен быть вложенными?
165. Для чего применяются алиасы пространства имен?
166. Как сделать члены пространства имен доступными в нескольких (в пределе — во всех) файлах программного проекта?
167. Объясните разницу между статической и динамической инициализацией.
168. В чем состоит проблема инициализации глобальных статических переменных?
169. Какие элементы класса можно объявлять статическими?
170. Можно ли объявить в классе статическую константу? А константный статический массив?
171. А какие статические поля можно инициализировать непосредственно в классе?
172. Как определяются статические поля? В какой момент работы программы выполняется инициализация статических полей?
173. Сколько места в классе занимают статические поля ?

174. Чем отличается статический метод от обычного?
175. Какие методы класса не могут быть статическими?
176. Какие применения статических полей вы можете привести? А каким образом применяются статические методы?
177. Приведите структуру и принцип действия паттерна Singleton.
178. Для чего предназначены шаблоны?
179. Какие виды шаблонов в C++ вы знаете?
180. Объясните термин «инстанцирование шаблона».
181. В чем разница между определением и объявлением шаблона?
182. Объясните назначение ключевого слова `typename`.
183. Какие виды параметров разрешается задавать в шаблоне класса? А в шаблоне функции?
184. Можно ли параметрам шаблона присваивать значения по умолчанию?
185. Может ли параметром шаблона быть другой шаблон? Каковы особенности объявления параметра-шаблона?
186. Что такое специализация шаблона? Объясните разницу между полной и частичной специализацией.
187. Разрешается ли специализировать шаблон функции?
188. Может ли класс-шаблон быть вложенным в другой класс-шаблон? А в обычный класс?
189. Можно ли объявить в классе шаблонный метод? А шаблонный конструктор?
190. Можно ли перегружать функцию-шаблон?
191. Какие параметры функции-шаблона выводятся автоматически?
192. Может ли шаблон класса быть наследником обычного класса? А обычный класс от шаблона?
193. Объясните, что такое класс свойств (класс трактовок).
194. Каким образом можно использовать возможность наследования обычного класса от шаблона?
195. Может ли шаблонный конструктор быть конструктором по умолчанию?
196. Для чего применяются директивы явного инстанцирования?
197. Объясните, в чем состоят проблемы, возникающие при разделении шаблонного класса на интерфейс и реализацию?
198. Что такое «модель явного инстанцирования» и как она работает?
199. Может ли шаблонный класс иметь «друзей»?
200. Какие проблемы возникают при объявлении дружественной функции для класса-шаблона?
201. Разрешается ли определять в классе-шаблоне статические поля? А статические методы?
202. Что такое «инициализация нулем»?
203. Что является единицей памяти в C++? Какие требования к размеру единицы памяти прописаны в стандарте C++?
204. В каких единицах выдает результат операция `sizeof`? Какие типы данных имеют размер 1?
205. Какие три вида памяти входят в модель памяти C++?
206. Сколько видов динамической памяти обеспечивает C++?
207. Какие функции для работы с динамической памятью достались C++ по наследству от C? В какую библиотеку они включены?
208. Какие функции выделяют память, и с помощью каких функций память освобождается?
209. Какое важное отличие имеет функция `calloc()` от функции `malloc()`?
210. Какие действия выполняют функции выделения памяти, если память не может быть выделена?

211. Зависит ли объем выделенной памяти от типа указателя? Влияет ли выравнивание на объем выделяемой динамической памяти?
212. Можно ли с помощью функции `realloc()` уменьшить объем выделенной памяти?
213. Что произойдет, если функции `free()` передать в качестве аргумента нулевой указатель?
214. В чем главное отличие объектно-ориентированного механизма `new/delete` от механизма `malloc()/free()`?
215. Сколько существует форм `new/delete`? В чем их отличие?
216. Какие типы являются POD-типами? Чем отличается работа механизма `new/delete` с POD-объектами и nonPOD-объектами?
217. Какие функции выполняет обработчик `new`?
218. Можно ли реализовать собственный обработчик `new` и «прицепить» его к механизму `new/delete`?
219. В чем главное отличие объединения от других видов классов C++?
220. Может ли объединение участвовать в иерархии наследования?
221. Разрешается ли определять для объединения конструкторы и деструктор? А виртуальные функции?
222. В чем похожи и чем отличаются объединение и размещающий `new`?
223. Объясните, почему при использовании размещающего `new` нужно явным образом вызывать деструктор?
224. Зачем нужны интеллектуальные указатели?
225. Что такое «стратегия владения»? Сколько стратегий владения вы знаете?
226. Какой интеллектуальный указатель реализован в стандартной библиотеке STL, и какую стратегию владения он реализует?
227. Объясните, в чем преимущества и недостатки интеллектуальных указателей со счетчиком ссылок.
228. Разрешается ли перегружать `new` и `delete` и какими способами?
229. Опишите схему функции, перегружающей глобальную функцию `new`.
230. Отличается ли реализация перегруженной функции `new[]()` для массивов от реализации «обычной» функции `new()`?
231. Как вы думаете, почему функции `new/delete`, перегружаемые для класса, являются статическими?
232. Зачем при перегрузке `new/delete` для класса нужно проверять размер запрашиваемой памяти?
233. Объясните, чем определяется «динамичность» контейнеров?
234. Что такое «стратегия распределения памяти», и какие стратегии выделения памяти вы знаете?
235. Рассмотрите следующую стратегию распределения памяти: память выделяется для нескольких элементов блоками фиксированной длины, но блоки связываются в список. Для какого вида контейнера можно использовать такую стратегию?
236. Какие операции можно перегрузить для доступа к элементам двумерного массива?
237. В чем заключаются сложности использования операции индексирования `[]` для доступа к элементам двумерного массива?
238. Каковы способы реализации операций с контейнерами?
239. Какую конструкцию можно назвать «обобщенный алгоритм»?
240. Каким образом объявить указатель на метод?
241. Объясните разницу между указателем на функцию и указателем на метод.
242. Каким образом получить адрес метода?
243. Можно ли указателю на функцию присваивать адрес метода?
244. Какие операции определены в C++ для косвенного вызова метода через указатель?
245. Что такое «функтор»? Приведите пример функционального класса.
246. Какими способами функтор вызывается?

247. Можно ли использовать наследование при разработке функторов?
248. Разрешается ли операцию вызова функции () определять как виртуальный метод? А как статический?
249. В чем преимущества функторов перед указателями на функции?
250. Объясните, зачем нужны адаптеры функторов? Какие виды адаптеров вы знаете?
251. Как используются классы свойств при разработке функторов?
252. Объясните, что такое «композиция» и приведите примеры?
253. Объясните, чем отличается множественное наследование от простого?
254. Приведите структуру и принцип действия паттерна Adapter.
255. Сформулируйте основную проблему множественного наследования.
256. Выполняется ли принцип подстановки при открытом множественном наследовании?
257. Что такое виртуальное наследование? Каковы его преимущества и недостатки по сравнению с обычным наследованием?
258. Может ли виртуальное наследование быть одиночным?
259. Влияет ли виртуальное наследование на размер класса?
260. Объясните, каким образом с помощью виртуального наследования можно вообще запретить наследование.
261. Какие средства C++ составляют RTTI?
262. Объясните разницу между повышающим, понижающим и перекрестным приведением.
263. Какими свойствами должен обладать класс, чтобы с ним работал механизм RTTI?
264. В чем приведение указателей отличается от приведения ссылок?
265. Какие исключения связаны с механизмом RTTI?
266. Что такое «поток» — дайте определение.
267. Как классифицируются потоки, реализованные в библиотеках ввода/вывода C++?
268. Что такое буферизация и зачем она нужна?
269. Какие библиотеки ввода/вывода реализованы в C++ и чем они отличаются?
270. Перечислите стандартные потоки и объясните их назначение.
271. Зачем нужен процесс форматирования и когда он выполняется?
272. Что такое «форматная строка», и в каких функциях она используется?
273. Объясните назначение элементов спецификатора формата.
274. Сколько спецификаторов формата может быть в форматной строке?
275. Какой из элементов спецификатора формата не является умалчиваемым?
276. Перечислите несколько известных вам обозначений типов в спецификаторе формата, и укажите их назначение.
277. Сколько модификаторов типа вы знаете, и какую роль модификатор типа играет в спецификаторе формата?
278. С помощью какого флага можно выровнять выводимое значение влево? А каким образом вывести ведущие нули?
279. Какое действие оказывают на выводимую строку ширина, точность и флаги в спецификаторе формата?
280. Для чего в спецификаторе формата может использоваться символ звездочка («*»)? Чем отличается действие этого символа при вводе и при выводе?
281. Каковы особенности ввода строк?
282. Каким образом ограничить набор вводимых символов при вводе?
283. Что является главной проблемой при использовании форматного ввода/вывода из библиотеки <cstdio>?
284. Объясните, для чего нужны строковые потоки. Почему строковые потоки — всегда форматируемые?
285. С помощью каких функций выполняется работа со строковыми потоками?
286. Можно ли использовать тип string (и каким образом) со строковыми потоками?
287. Объясните, в чем заключается различие между текстовым и двоичным файлом.

288. Объясните, что означает «открыть» файл и «закрыть» файл?
289. Каким образом внешний файл связывается с потоком?
290. Можно ли один и тот же поток связать с разными файлами? А один и тот же файл с разными потоками?
291. Перечислите режимы открытия файла. Чем отличается режим “r” от режима “a”?
292. Какую роль в режиме открытия играет знак плюс («+»)?
293. В каких случаях необходимо следить за ситуацией «конец файла»? Каким способом это делается?
294. Можно ли текстовый файл открыть как двоичный? А двоичный — как текстовый?
295. Какие функции ввода/вывода используются для обмена с текстовыми файлами?
296. Перечислите функции ввода/вывода для работы с двоичными файлами.
297. Какие функции реализованы в библиотеке <stdio> для обеспечения прямого доступа к записям двоичного файла? Можно ли их использовать для работы с текстовыми файлами?
298. Объясните назначение функции fseek().
299. Чем отличается функция ftell() от функции fgetpos()?
300. Объясните, что означает «перенаправление» потока? Какие потоки можно перенаправлять и куда?
301. Каким образом перенаправление ввода можно использовать для ввода строк с пробелами?
302. В чем преимущества объектно-ориентированной библиотеки по сравнению с процедурной?
303. В каких состояниях может находиться поток? Каким образом отслеживается состояние «конец потока»?
304. Какие объектно-ориентированные потоки связаны со стандартными потоками?
305. Чем отличаются объектно-ориентированные строковые потоки от процедурных строковых потоков?
306. Каким образом строковые потоки можно использовать для ограничения ширины поля ввода? А можно ли с той же целью использовать строковые потоки <stdio>?
307. Сравните средства форматирования объектно-ориентированной и процедурной библиотеки.
308. Каким образом ввести строку типа string с пробелами?
309. Каково назначение флагов форматирования? Какие средства реализованы в библиотеке для работы с флагами форматирования?
310. Что такое «манипулятор»? В чем преимущества манипуляторов перед флагами форматирования?
311. Как связываются файлы с потоками в объектно-ориентированной библиотеке?
312. Можно ли файлы, записанные функциями библиотеки <stdio>, прочитать объектно-ориентированными средствами? А наоборот?
313. Перечислите режимы открытия объектно-ориентированных файловых потоков. каким образом комбинируются режимы открытия файловых потоков?
314. Обязательно ли закрывать файл, связанный с объектно-ориентированным файловым потоком? А открывать?
315. Каким образом открыть файловый поток для чтения и записи одновременно?
316. Как открыть файловый поток для дозаписи?
317. Можно ли вывести значение переменной в двоичном виде и как это сделать?
318. Разрешается ли наследовать от классов библиотеки ввода/вывода?
319. Каким образом можно перенаправить объектно-ориентированный поток?
320. Как используется буфер потока для копирования потока?
321. Какими операциями выполняется форматированный ввод/вывод в файловые потоки? А неформатированный?
322. Реализованы ли в объектно-ориентированной библиотеке средства прямого доступа к

файловым потокам? Сравните их с аналогичными средствами библиотеки `<cstdio>`.

323. С какими объектно-ориентированными потоками разрешается, и с какими не разрешается использовать средства прямого доступа?
324. Покажите, каким образом можно выполнить перегрузку операций ввода/вывода для нового типа данных.
325. Как выполняется обработка ошибок ввода/вывода в объектно-ориентированной библиотеке?
326. Какое стандартное исключение генерируется при ошибках ввода/вывода? Обязательно ли оно генерируется?
327. Чем стандартные широкие потоки отличаются от узких?
328. Что такое — «локаль», и каково ее назначение?
329. Как установить русский шрифт при выводе в консольное окно?
330. Чем отличается ли ввод/вывод широких файловых потоков от узких?
331. Перечислите все последовательные контейнеры стандартной библиотеки. Чем они отличаются друг от друга?
332. Перечислите адаптеры последовательных контейнеров и дайте их подробную характеристику.
333. Почему для адаптеров-очередей нельзя использовать вектор в качестве базового?
334. Чем простая очередь `queue` отличается от приоритетной очереди `priority_queue`?
335. Каким требованиям должны удовлетворять элементы контейнера?
336. Могут ли быть указатели элементами контейнера? А итераторы?
337. Почему нельзя использовать в качестве элементов контейнера стандартный интеллектуальный указатель `auto_ptr`?
338. Зачем в контейнере `list` реализованы собственные методы сортировки поиска и слияния? Можно ли пользоваться соответствующими стандартными алгоритмами при обработке списка?
339. Перечислите типовые виды конструкторов, с помощью которых можно создавать последовательный контейнер.
340. Можно ли инициализировать контейнер элементами встроеного массива? А элементами другого контейнера? Какими способами это можно сделать?
341. Почему конструктор инициализации, параметрами которого являются итераторы, сделан шаблонным во всех контейнерах?
342. Какие методы реализованы в контейнере-векторе для доступа к элементам?
343. Отличается ли функция `at()` доступа по индексу от перегруженной операции индексирования и чем?
344. Перечислите методы контейнера `deque`, относящиеся к определению размеров контейнера.
345. Чем метод `size()` отличается от метода `capacity()`? А в чем отличие этих методов от метода `max_size()`?
346. Перечислите методы контейнера `list`, предназначенные для вставки удаления и замены элементов. Отличаются ли эти методы от соответствующих методов вектора и дека?
347. Каким образом выполняются операции сравнения контейнеров?
348. Разрешается ли изменять элемент ассоциативного контейнера, доступный в данный момент по итератору?
349. Какие контейнеры называются ассоциативными и почему?
350. Чем контейнер `map` отличается от контейнера `multimap`?

4.Обновление основной и дополнительной литературы (2018 г.)

В раздел 6. Учебно-методическое и информационное обеспечение дисциплины вносятся следующие изменения:

Дополнить раздел **Основная литература**

1. Алымова Елена Владимировна. Конечные автоматы и формальные языки: Учебник; ВО - Бакалавриат. - Ростов-на-Дону: Издательство Южного федерального университета (ЮФУ), 2018. - 292 с. - ВО - Бакалавриат. - ISBN 9785927523979. Ссылка на ресурс: <http://new.znaniium.com/go.php?id=1020503>

2. Бильфельд Николай Валентинович. Современные средства реализации автоматизированных систем. Работа с Google таблицами : Учебное пособие. - 1. - Москва ; Москва : Издательский Центр РИОР: ООО "Научно-издательский центр ИНФРА-М", 2018. - 171 с. - ISBN 9785369017210.

5. Состав программного обеспечения (ПО), современных профессиональных баз данных (БД) и информационно-справочные систем (ИСС) (2018 г.)

Перечень ПО

Таблица 1

| №п/п | Наименование ПО | Производитель | Способ распространения (лицензионное или свободно распространяемое) |
|------|-----------------------------|------------------|--|
| 1 | Adobe Master Collection CS4 | Adobe | лицензионное |
| 2 | Microsoft Office 2010 | Microsoft | лицензионное |
| 3 | Windows 7 Pro | Microsoft | лицензионное |
| 4 | AutoCAD 2010 Student | Autodesk | свободно распространяемое |
| 5 | Archicad 21 Rus Student | Graphisoft | свободно распространяемое |
| 6 | SPSS Statistics 22 | IBM | лицензионное |
| 7 | Microsoft Share Point 2010 | Microsoft | лицензионное |
| 8 | SPSS Statistics 25 | IBM | лицензионное |
| 9 | Microsoft Office 2013 | Microsoft | лицензионное |
| 10 | ОС «Альт Образование» 8 | ООО «Базальт СПО | лицензионное |
| 11 | Microsoft Office 2013 | Microsoft | лицензионное |
| 12 | Windows 10 Pro | Microsoft | лицензионное |
| 13 | Kaspersky Endpoint Security | Kaspersky | лицензионное |

Перечень БД и ИСС

Таблица 2

| №п/п | Наименование |
|------|--|
| | Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2018 г. Web of Science Scopus |
| | Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2018 г. Журналы Cambridge University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis |

| | |
|--|--|
| | Электронные издания издательства Springer |
| | Профессиональные полнотекстовые БД JSTOR Издания по общественным и гуманитарным наукам |
| | Компьютерные справочные правовые системы Консультант Плюс, Гарант |

Составитель:

Старший преподаватель кафедры информационных технологий и систем, Е.П. Охапкина

6. Обновление структуры дисциплины (модуля) для очной формы обучения (2019 г.)**Структура дисциплины (модуля) для очной формы обучения**

Общая трудоёмкость дисциплины составляет 5 з.е., 180 ч., в том числе контактная работа обучающихся с преподавателем 70 ч., самостоятельная работа обучающихся 92 ч., промежуточная аттестация – 18 ч.

| № п/п | Раздел дисциплины/темы | Семестр | Виды учебной работы (в часах) | | | | | Самостоятельная работа | Формы текущего контроля успеваемости, форма промежуточной аттестации (<i>по семестрам</i>) |
|------------------|--|---------|-------------------------------|---------|----------------------|----------------------|--------------------------|------------------------|--|
| | | | контактная | | | | | | |
| | | | Лекции | Семинар | Практические занятия | Лабораторные занятия | Промежуточная аттестация | | |
| 2 семестр | | | | | | | | | |
| 1 | Раздел 1. Программирование реляционной базы данных на языке C++ | 2 | 4 | | 4 | | | 22 | Защита отчета по лабораторной работе №1 |
| 2 | Раздел 2. Основы программирования на языке Python. | 2 | 6 | | 6 | | | 22 | Защита отчета по лабораторной работе №2 |
| 3 | Раздел 3. Алгоритмы анализа данных с использованием библиотеки Scikit-Learn. | 2 | 10 | | 10 | | | 22 | Защита отчета по лабораторной работе №3 |
| | зачет | | | | | | | | Зачет по билетам |
| | итого: | | 20 | | 22 | | | 66 | |
| 3 семестр | | | | | | | | | |
| 4 | Раздел 4. Программируемая библиотека TensorFlow. | 3 | 6 | | 8 | | | 12 | Защита отчета по лабораторной работе №4 |
| 5 | Раздел 5. Нейросетевое моделирование на языке Python. | 3 | 6 | | 8 | | | 14 | Защита отчета; по лабораторной работе №5 |
| | <i>Экзамен</i> | 3 | | | | | 18 | | Экзамен по билетам |
| | итого: | | 12 | | 16 | | 18 | 26 | |
| | Всего | | 32 | | 38 | | 18 | 92 | |

7. Обновление основной и дополнительной литературы (2019 г.)

В раздел **6. Учебно-методическое и информационное обеспечение дисциплины** вносятся следующие изменения:

Дополнить раздел *Дополнительная литература*

1. Гниденко Ирина Геннадиевна. Технологии и методы программирования : Учебное пособие / И. Г. Гниденко [и др.]. - Электрон. дан. - Москва : Издательство Юрайт, 2019. - 235. - (Бакалавр. Прикладной курс). - Режим доступа: <https://www.biblio-online.ru>. - ISBN 978-5-534-02816-4 : 479.00.
2. Федоров Дмитрий Юрьевич. Программирование на языке высокого уровня python : Учебное пособие / Д. Ю. Федоров. - 2-е изд. - Электрон. дан. - Москва : Издательство Юрайт, 2019. - 161. - (Бакалавр. Прикладной курс). - Режим доступа: <https://www.biblio-online.ru>. - ISBN 978-5-534-10971-9 : 349.00.

8. Состав программного обеспечения (ПО), современных профессиональных баз данных (БД) и информационно-справочные систем (ИСС) (2019 г.)

Перечень ПО

| №п /п | Наименование ПО | Производитель | Способ распространения (лицензионное или свободно распространяемое) |
|-------|-----------------------------|------------------|--|
| 1 | Adobe Master Collection CS4 | Adobe | лицензионное |
| 2 | Microsoft Office 2010 | Microsoft | лицензионное |
| 3 | Windows 7 Pro | Microsoft | лицензионное |
| 4 | AutoCAD 2010 Student | Autodesk | свободно распространяемое |
| 5 | Archicad 21 Rus Student | Graphisoft | свободно распространяемое |
| 6 | SPSS Statistics 22 | IBM | лицензионное |
| 7 | Microsoft Share Point 2010 | Microsoft | лицензионное |
| 8 | SPSS Statistics 25 | IBM | лицензионное |
| 9 | Microsoft Office 2013 | Microsoft | лицензионное |
| 10 | ОС «Альт Образование» 8 | ООО «Базальт СПО | лицензионное |
| 11 | Microsoft Office 2013 | Microsoft | лицензионное |
| 12 | Windows 10 Pro | Microsoft | лицензионное |
| 13 | Kaspersky Endpoint Security | Kaspersky | лицензионное |
| 14 | Microsoft Office 2016 | Microsoft | лицензионное |
| 15 | Visual Studio 2019 | Microsoft | лицензионное |
| 16 | Adobe Creative Cloud | Adobe | лицензионное |

1. Перечень БД и ИСС

| №п /п | Наименование |
|-------|--|
| 1 | Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2019 г. Web of Science |

| | Scopus |
|---|--|
| 2 | Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2019 г. Журналы Cambridge University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis |
| 3 | Профессиональные полнотекстовые БД JSTOR Издания по общественным и гуманитарным наукам Электронная библиотека Grebennikon.ru |
| 4 | Компьютерные справочные правовые системы Консультант Плюс, Гарант |

Составитель:

Старший преподаватель кафедры информационных технологий и систем, Е.П. Охупкина

9. Обновление структуры дисциплины (модуля) для очной формы обучения (2020 г.)**2. Структура дисциплины (модуля) для очной формы обучения**

Общая трудоёмкость дисциплины составляет 5 з.е., 190 ч., в том числе контактная работа обучающихся с преподавателем 70 ч., самостоятельная работа обучающихся 102 ч., промежуточная аттестация – 18 ч.

| № п/п | Раздел дисциплины/темы | Семестр | Виды учебной работы (в часах) | | | | | Самостоятельная работа | Формы текущего контроля успеваемости, форма промежуточной аттестации (<i>по семестрам</i>) |
|------------------|--|---------|----------------------------------|---------|----------------------|----------------------|--------------------------|------------------------|--|
| | | | контактная | | | | | | |
| | | | Лекции | Семинар | Практические занятия | Лабораторные занятия | Промежуточная аттестация | | |
| 2 семестр | | | | | | | | | |
| 1 | Раздел 1. Программирование реляционной базы данных на языке C++ | 2 | 4 | | 4 | | | 24 | Защита отчета по лабораторной работе №1 |
| 2 | Раздел 2. Основы программирования на языке Python. | 2 | 6 | | 6 | | | 24 | Защита отчета по лабораторной работе №2 |
| 3 | Раздел 3. Алгоритмы анализа данных с использованием библиотеки Scikit-Learn. | 2 | 10 | | 10 | | | 24 | Защита отчета по лабораторной работе №3 |
| | зачет | | | | | | | | Зачет по билетам |
| | итого: | | 20 | | 22 | | | 72 | |
| 3 семестр | | | | | | | | | |
| 4 | Раздел 4. Программируемая библиотека TensorFlow. | 3 | 6 | | 8 | | | 12 | Защита отчета по лабораторной работе №4 |
| 5 | Раздел 5. Нейросетевое моделирование на языке Python. | 3 | 6 | | 8 | | | 16 | Защита отчета; по лабораторной работе №5 |
| | <i>Экзамен</i> | 3 | | | | | 18 | | Экзамен по билетам |
| | итого: | | 12 | | 16 | | 18 | 30 | |
| | Всего | | 32 | | 38 | | 18 | 102 | |

10. Обновление основной и дополнительной литературы (2020 г.)

В раздел **6. Учебно-методическое и информационное обеспечение дисциплины** вносятся следующие изменения:

2. Дополнить раздел Основная литература
 1. Колдаев Виктор Дмитриевич. Численные методы и программирование : Учебное пособие; / Московский институт электронной техники. - 1. - Москва : Издательский Дом "ФОРУМ", 2020. - 336 с. - ISBN 9785819907795. Ссылка на ресурс: <http://new.znanium.com/go.php?id=1041477>
 2. Гуриков Сергей Ростиславович. Основы алгоритмизации и программирования на Python : Учебное пособие; / Московский технический университет связи и информатики. - 1. - Москва : Издательство "ФОРУМ", 2020. - 343 с. - ISBN 9785000915530. Ссылка на ресурс: <http://new.znanium.com/go.php?id=1042452>
 3. Жуков Роман Александрович. Язык программирования Python: практикум : Учебное пособие; / Финансовый университет при Правительстве Российской Федерации, Тульский ф-л. - 1. - Москва : ООО "Научно-издательский центр ИНФРА-М", 2020. - 216 с. - ISBN 9785160156385. <http://new.znanium.com/go.php?id=1044193>

11. В элемент рабочей программы **п.4 Образовательные технологии** вносятся следующие изменения:

В период временного приостановления посещения обучающимися помещений и территории РГГУ. для организации учебного процесса с применением электронного обучения и дистанционных образовательных технологий могут быть использованы следующие образовательные технологии:

- видео-лекции;
- онлайн-лекции в режиме реального времени;
- электронные учебники, учебные пособия, научные издания в электронном виде и доступ к иным электронным образовательным ресурсам;
- системы для электронного тестирования;
- консультации с использованием телекоммуникационных средств.

12. В элемент рабочей программы **7. Материально-техническое обеспечение дисциплины/модуля** вносятся следующие изменения:

Перечень БД и ИСС

| №п/п | Наименование |
|------|--|
| 1 | Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2020 г. Web of Science Scopus |
| 2 | Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2020 г. Журналы Cambridge University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis |
| 3 | Профессиональные полнотекстовые БД JSTOR |

| | |
|---|--|
| | Издания по общественным и гуманитарным наукам Электронная библиотека Grebennikon.ru |
| 4 | Компьютерные справочные правовые системы Консультант Плюс, Гарант |

В элемент рабочей программы **7. Материально-техническое обеспечение дисциплины/модуля** вносятся следующие изменения:

Состав программного обеспечения (ПО)

| №п /п | Наименование ПО | Производитель | Способ распространения (лицензионное или свободно распространяемое) |
|-------|-----------------------------|------------------|--|
| 1 | Adobe Master Collection CS4 | Adobe | лицензионное |
| 2 | Microsoft Office 2010 | Microsoft | лицензионное |
| 3 | Windows 7 Pro | Microsoft | лицензионное |
| 4 | AutoCAD 2010 Student | Autodesk | свободно распространяемое |
| 5 | Archicad 21 Rus Student | Graphisoft | свободно распространяемое |
| 6 | SPSS Statistics 22 | IBM | лицензионное |
| 7 | Microsoft Share Point 2010 | Microsoft | лицензионное |
| 8 | SPSS Statistics 25 | IBM | лицензионное |
| 9 | Microsoft Office 2013 | Microsoft | лицензионное |
| 10 | ОС «Альт Образование» 8 | ООО «Базальт СПО | лицензионное |
| 11 | Microsoft Office 2013 | Microsoft | лицензионное |
| 12 | Windows 10 Pro | Microsoft | лицензионное |
| 13 | Kaspersky Endpoint Security | Kaspersky | лицензионное |
| 14 | Microsoft Office 2016 | Microsoft | лицензионное |
| 15 | Visual Studio 2019 | Microsoft | лицензионное |
| 16 | Adobe Creative Cloud | Adobe | лицензионное |
| 17 | Zoom | Zoom | лицензионное |

Составитель:

Старший преподаватель кафедры информационных технологий и систем, Е.П. Охупкина