

Министерство науки и высшего образования Российской Федерации
Министерство образования и науки Кыргызской Республики

Государственное образовательное учреждение высшего профессионального
образования Кыргызско-Российский Славянский университет имени первого
президента Российской Федерации Б.Н. Ельцина

Естественно-технический факультет
Кафедра «Сети связи и системы коммуникации»

УТВЕРЖДАЮ
Зав. кафедры «Сети связи и системы
коммуникации»




 М. Оконов

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущей и промежуточной аттестации

(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)

По дисциплине	Программное обеспечение инфокоммуникационных технологий
Направление	<u>11.03.02 Инфокоммуникационные технологии и системы связи</u>
Профиль	<u>Сети связи и системы коммутации</u>
Квалификация выпускника	<u>Бакалавр</u>
Форма обучения	очная
Учебный план	<u>2022 год</u>

Предисловие

1. Назначение: Фонд оценочных средств по дисциплине Программное обеспечение инфокоммуникационных технологий предназначен для формирования у студентов направления 11.03.02 Инфокоммуникационные технологии и системы связи общепрофессиональных компетенций ОПК-3, ОПК-4.
2. Фонд оценочных средств текущей и промежуточной аттестации на основе рабочей программы дисциплины Программное обеспечение инфокоммуникационных технологий в соответствии с образовательной программой по направления 11.03.02 Инфокоммуникационные технологии и системы связи.
3. Разработчик Мамадалиева Ж.Б., Ст. препод. 
4. ФОС рассмотрен и утвержден на заседании выпускающей кафедры «Сети связи и системы коммуникации»
5. Проведена экспертиза ФОС. Члены экспертной группы, проводившие внутреннюю экспертизу:
Председатель Оконов М., доцент к.т.н., зав. кафедры «Сети связи и системы коммуникации»;
Члены экспертной группы:
Сагымбаев А.А., д.т.н., кафедры «Сети связи и системы коммуникации» 
Джылышбаева М.Н., доцент к.т.н., кафедры «Сети связи и системы коммуникации»


Экспертное заключение: фонд оценочных средств соответствует ОП ВО по направлению подготовки 11.03.02 Инфокоммуникационные технологии и системы связи, рекомендуется для оценивания уровня сформированности компетенций при проведении текущего контроля успеваемости и промежуточной аттестации студентов по дисциплине «Программное обеспечение инфокоммуникационных технологий».

Срок действия ФОС: на срок реализации образовательной программы.

Паспорт фонда оценочных средств

для проведения текущей и промежуточной аттестации

По дисциплине Направление Профиль Квалификация выпускника Форма обучения Учебный план	<u>Программное обеспечение</u> <u>инфокоммуникационных технологий</u> <u>11.03.02 Инфокоммуникационные технологии и</u> <u>системы связи</u> <u>Сети связи и системы коммутации</u> <u>Бакалавр</u> очная <u>2022 год</u>
--	--

Код оцениваемой компетенции (или её части)	Модуль, раздел, тема (в соответствии с Программой)	Тип контроля	Вид контроля	Компонент фонда оценочных средств	Количество заданий для каждого уровня, шт.	
					Базовый	Повышенный
ОПК-3, ОПК-4	Тема 1-9	текущий	устный	Вопросы для собеседования по лабораторным работам	113	25
ОПК-3, ОПК-4	Тема 1-9	промежуточный	устный	Вопросы к зачету	62	34

Вопросы к зачету

по дисциплине

Программное обеспечение инфокоммуникационных технологий

Базовый уровень

1. Что такое система контроля версий (СКВ) и каково ее назначение? В чем недостатки локальных и централизованных СКВ?
2. К какой системе контроля версий (СКВ) относится Git? В чем концептуальное отличие Git от других СКВ?
3. Как обеспечивается целостность хранимых данных в Git?
4. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?
5. Что такое профиль пользователя в GitHub?
6. Какие бывают репозитории в GitHub?
7. Укажите основные этапы модели работы с GitHub.
8. Как осуществляется первоначальная настройка Git после установки?
9. Опишите этапы создания репозитория в GitHub.
10. Какие типы лицензий поддерживаются GitHub при создании репозитория?
11. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?
12. Как проверить состояние локального репозитория Git?
13. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/измененного файла под версионный контроль с помощью команды `*git add*`; фиксации (коммита) изменений с помощью команды `*git commit*` и отправки изменений на сервер с помощью команды `*git push*`?
14. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии.
15. Опишите основные этапы установки Python в Windows и Linux.
16. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?
17. Как осуществить проверку работоспособности пакета Anaconda?
18. Как задать используемый интерпретатор языка Python в IDE PyCharm?
19. Как осуществить запуск программы с помощью IDE PyCharm?
20. В чем суть интерактивного и пакетного режимов работы Python?
21. Почему язык программирования Python называется языком динамической типизации?
22. Какие существуют основные типы в языке программирования Python?
23. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?
24. Как получить список ключевых слов в Python?
25. Каково назначение функций `_id()` и `_type()`?

26. Что такое изменяемые и неизменяемые типы в Python.
27. Чем отличаются операции деления и целочисленного деления?
28. Какие имеются средства в языке Python для работы с комплексными числами?
29. Каково назначение и основные функций библиотек (модулей) `_math_` и `*cmath*`?
30. Каково назначение именованных параметров `*sep*` и `*end*` в функции `_print()`?
31. Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?
32. Для чего нужны диаграммы деятельности UML?
33. Что такое состояние действия и состояние деятельности?
34. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?
35. Какой алгоритм является алгоритмом разветвляющейся структуры?
36. Чем отличается разветвляющийся алгоритм от линейного?
37. Что такое условный оператор? Какие существуют его формы?
38. Какие операторы сравнения используются в Python?
39. Что называется простым условием? Приведите примеры.
40. Что такое составное условие? Приведите примеры.
41. Какие логические операторы допускаются при составлении сложных условий?
42. Может ли оператор ветвления содержать внутри себя другие ветвления?
43. Какой алгоритм является алгоритмом циклической структуры?
44. Какие поддерживаются типы циклов в языке Python?
45. Назовите назначение и способы применения функции `*range*`. Как с помощью функции `*range*` организовать перебор значений от 15 до 0 с шагом 2?
46. Могут ли быть циклы вложенными? Как образуется бесконечный цикл и как выйти из него?
47. Для чего нужен оператор `*break*`? Где употребляется оператор `*continue*` и для чего он используется?
48. Для чего нужны стандартные потоки `*stdout*` и `*stderr*`? Как в Python организовать вывод в стандартный поток `*stderr*`?
49. Каково назначение функции `*exit*`?
50. 50. Что такое строки в языке Python? Какие существуют способы задания строковых литералов в языке Python?
51. Какие операции и функции существуют для строк? Почему строки Python относятся к неизменяемому типу данных?
52. Как осуществляется индексирование строк?
53. Как осуществляется работа со срезами для строк?
54. Как проверить то, что каждое слово в строке начинается с заглавной буквы?
55. Как проверить строку на вхождение в неё другой строки?
56. Как найти индекс первого вхождения подстроки в строку?
57. Как подсчитать количество символов в строке? Как найти подстроку в заданной части строки?
58. Как подсчитать то, сколько раз определённый символ встречается в строке?
59. Форматирование строк в Python. Что такое f-строки и как ими пользоваться? В чем преимущества использования метода `*format()*`?
60. Как в заданной строке заменить на что-либо все вхождения некоей подстроки?
61. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?
62. Как привести к верхнему регистру первый символ каждого слова в строке?

Продвинутый уровень

1. Как пользоваться методом `*partition()*`? В каких ситуациях пользуются методом `*rfind()*`?

2. Что такое списки в языке Python? Как осуществляется создание списка в Python?
3. Как организовано хранение списков в оперативной памяти?
4. Каким образом можно перебрать все элементы списка?
5. Какие существуют арифметические операции со списками?
6. Как проверить есть ли элемент в списке? Как выполнить сортировку списка?
7. Как определить число вхождений заданного элемента в списке?
8. Как осуществляется добавление (вставка) элемента в список?
9. Как удалить один или несколько элементов из списка?
10. Что такое списковое включение и как с его помощью осуществлять обработку списков?
11. Как осуществляется доступ к элементам списков с помощью срезов?
12. Какие существуют функции агрегации для работы со списками?
13. Каково назначение функции `*sorted*` в языке Python. В чем ее отличие от метода `*sort*` списков?
14. Что такое словари в языке Python? Какие способы создания словарей Вам известны?
15. Может ли функция `*len()*` быть использована при работе со словарями? Какие методы обхода словарей Вам известны?
16. Какими способами можно получить значения из словаря по ключу? Какими способами можно установить значение в словаре по ключу?
17. Что такое словарь включений?
18. Каково назначение функции `*zip()*`? Приведите примеры ее использования.
19. Каково назначение модуля `*datetime*`? Каким функционалом по работе с датой и временем обладает этот модуль?
20. Каково назначение функций в языке программирования Python?
21. Как определяется функция в языке Python? Каково назначение операторов `*def*` и `*return*`?
22. Каково назначение локальных и глобальных переменных при написании функций в Python?
23. Как вернуть несколько значений из функции Python?
24. Какие существуют способы передачи значений в функцию?
25. Как задать значение аргументов функции по умолчанию?
26. Каково назначение `lambda`-выражений в языке Python?
27. Как осуществляется объявление класса в языке Python?
28. Чем атрибуты класса отличаются от атрибутов экземпляра? Каково назначение методов класса?
29. Для чего предназначен метод ``__init__`()` класса?
30. Каково назначение `*self*`?
31. В чем отличие между атрибутами класса и объекта? Как добавить атрибуты в класс?
32. Как осуществляется управление доступом к методам и атрибутам в языке Python?
33. Что такое наследование и полиморфизм и как они реализованы в языке Python?
34. Каково назначение функции `*isinstance*`?

1. Критерии оценивания компетенций

оценка «отлично» выставляется, если студент продемонстрировал высокое умение применять полученные знания на практике через решение конкретной задачи по применению информационных технологий, свободно без затруднений справился с поставленной задачей, показав владение разносторонними приемами и навыками ее выполнения, не допустил ошибок и неточностей;

оценка «хорошо» выставляется, если студент продемонстрировал умение применять полученные знания на практике через решение конкретной задачи по применению информационных технологий, справился с поставленной задачей, показав владение необходимыми приемами и навыками ее выполнения, при этом допустил не более одной ошибки;

оценка «удовлетворительно» выставляется, если студент продемонстрировал посредственное умение применять полученные знания на практике через решение конкретной задачи по применению информационных технологий, с трудом справился с поставленной задачей, при этом допустил не более двух ошибок;

оценка «неудовлетворительно» выставляется, если студент не продемонстрировал умение применять полученные знания на практике через решение конкретной задачи по применению информационных технологий, не справился с поставленной задачей или допустил при ее решении три и более серьезные ошибки.

2. Описание шкалы оценивания

Промежуточная аттестация в форме экзамена предусматривает проведение обязательной экзаменационной процедуры и оценивается 40 баллами из 100. Минимальное количество баллов, необходимое для допуска к экзамену, составляет 33 балла. Положительный ответ студента на экзамене оценивается рейтинговыми баллами в диапазоне от **20** до **40** ($20 \leq S_{\text{ЭКЗ}} \leq 40$), оценка **меньше 20** баллов считается неудовлетворительной.

Шкала соответствия рейтингового балла экзамена 5-балльной системе

Рейтинговый балл по дисциплине	Оценка по 5-балльной системе
35 – 40	Отлично
28 – 34	Хорошо
20 – 27	Удовлетворительно

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедура проведения экзамена осуществляется в соответствии с Положением о
Процедура проведения экзамена осуществляется в соответствии с Положением о проведении
текущего контроля успеваемости и промежуточной аттестации обучающихся по
образовательным программам высшего образования в СКФУ.

В экзаменационный билет включаются три вопроса, два из которых относятся к
базовому уровню, один к повышенному.

Для подготовки по билету отводится 40 минут.

При подготовке к ответу студенту предоставляется право пользования рабочей
программой дисциплины и фондом оценочных средств.

Вопросы для собеседования по лабораторным работам

Базовый уровень

1. Что такое СКВ и каково ее назначение?
2. В чем недостатки локальных и централизованных СКВ?
3. К какой СКВ относится Git?
4. В чем концептуальное отличие Git от других СКВ?
5. Как обеспечивается целостность хранимых данных в Git?
6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?
7. Что такое профиль пользователя в GitHub?
8. Какие бывают репозитории в GitHub?
9. Укажите основные этапы модели работы с GitHub.
10. Как осуществляется первоначальная настройка Git после установки?
11. Опишите этапы создания репозитория в GitHub.
12. Какие типы лицензий поддерживаются GitHub при создании репозитория?
13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?
14. Как проверить состояние локального репозитория Git?
15. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/измененного файла под версионный контроль с помощью команды ``git add``; фиксации (коммита) изменений с помощью команды ``git commit`` и отправки изменений на сервер с помощью команды ``git push``?
16. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии. *Примечание*: описание необходимо начать с команды ``git clone``.
17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.
18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git? Приведите как реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.
19. Опишите основные этапы установки Python в Windows и Linux.
20. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?
21. Как осуществить проверку работоспособности пакета Anaconda?
22. Как задать используемый интерпретатор языка Python в IDE PyCharm?
23. Как осуществить запуск программы с помощью IDE PyCharm?
24. В чем суть интерактивного и пакетного режимов работы Python?
25. Почему язык программирования Python называется языком динамической типизации?
26. Какие существуют основные типы в языке программирования Python?
27. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс

- объявления новых переменных и работа операции присваивания?
28. Как получить список ключевых слов в Python?
 29. Каково назначение функций `_id()` и `_type()`?
 30. Что такое изменяемые и неизменяемые типы в Python.
 31. Чем отличаются операции деления и целочисленного деления?
 32. Какие имеются средства в языке Python для работы с комплексными числами?
 33. Каково назначение и основные функции библиотеки (модуля) `_math`? По аналогии с модулем `_math` изучите самостоятельно назначение и основные функции модуля `_cmath`.
 34. Каково назначение именованных параметров ``sep`` и ``end`` в функции `_print()`?
 35. Каково назначение метода `_format()`? Какие еще существуют средства для форматирования строк в Python? Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python.
 36. Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?
 37. Для чего нужны диаграммы деятельности UML?
 38. Что такое состояние действия и состояние деятельности?
 39. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?
 40. Какой алгоритм является алгоритмом разветвляющейся структуры?
 41. Чем отличается разветвляющийся алгоритм от линейного?
 42. Что такое условный оператор? Какие существуют его формы?
 43. Какие операторы сравнения используются в Python?
 44. Что называется простым условием? Приведите примеры.
 45. Что такое составное условие? Приведите примеры.
 46. Какие логические операторы допускаются при составлении сложных условий?
 47. Может ли оператор ветвления содержать внутри себя другие ветвления?
 48. Какой алгоритм является алгоритмом циклической структуры?
 49. Типы циклов в языке Python.
 50. Назовите назначение и способы применения функции ``range``.
 51. Как с помощью функции ``range`` организовать перебор значений от 15 до 0 с шагом 2?
 52. Могут ли быть циклы вложенными?
 53. Как образуется бесконечный цикл и как выйти из него?
 54. Для чего нужен оператор ``break``?
 55. Где употребляется оператор ``continue`` и для чего он используется?
 56. Для чего нужны стандартные потоки `*stdout*` и `*stderr*`?
 57. Как в Python организовать вывод в стандартный поток `*stderr*`?
 58. Каково назначение функции ``exit``?
 59. Что такое строки в языке Python?
 60. Какие существуют способы задания строковых литералов в языке Python?
 61. Какие операции и функции существуют для строк?
 62. Как осуществляется индексирование строк?
 63. Как осуществляется работа со срезами для строк?x
 64. Почему строки Python относятся к неизменяемому типу данных?
 65. Как проверить то, что каждое слово в строке начинается с заглавной буквы?
 66. Как проверить строку на вхождение в неё другой строки?
 67. Как найти индекс первого вхождения подстроки в строку?
 68. Как подсчитать количество символов в строке?
 69. Как подсчитать то, сколько раз определённый символ встречается в строке?
 70. Что такое f-строки и как ими пользоваться?
 71. Как найти подстроку в заданной части строки?
 72. Как вставить содержимое переменной в строку, воспользовавшись методом `format()`?

73. Как узнать о том, что в строке содержатся только цифры?
74. Как разделить строку по заданному символу?
75. Как проверить строку на то, что она составлена только из строчных букв?
76. Как проверить то, что строка начинается со строчной буквы?
77. Можно ли в Python прибавить целое число к строке?
78. Как «перевернуть» строку?
79. Как объединить список строк в одну строку, элементы которой разделены дефисами?
80. Как привести всю строку к верхнему или нижнему регистру?
81. Как преобразовать первый и последний символы строки к верхнему регистру?
82. Как проверить строку на то, что она составлена только из прописных букв?
83. В какой ситуации вы воспользовались бы методом ``splitlines()```?
84. Как в заданной строке заменить на что-либо все вхождения некоей подстроки?
85. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?
86. Как узнать о том, что строка включает в себя только пробелы?
87. Что случится, если умножить некую строку на 3?
88. Как привести к верхнему регистру первый символ каждого слова в строке?
89. Как пользоваться методом ``partition()```?
90. В каких ситуациях пользуются методом ``rfind()```?
91. Что такое списки в языке Python?
92. Как осуществляется создание списка в Python?
93. Как организовано хранение списков в оперативной памяти?
94. Каким образом можно перебрать все элементы списка?
95. Какие существуют арифметические операции со списками?
96. Как проверить есть ли элемент в списке?
97. Как определить число вхождений заданного элемента в списке?
98. Как осуществляется добавление (вставка) элемента в список?
99. Как выполнить сортировку списка?
100. Как удалить один или несколько элементов из списка?
101. Что такое списковое включение и как с его помощью осуществлять обработку списков?
102. Как осуществляется доступ к элементам списков с помощью срезов?
103. Какие существуют функции агрегации для работы со списками?
104. Как создать копию списка?
105. Самостоятельно изучите функцию `*sorted*` языка Python. В чем ее отличие от метода `*sort*` списков?
106. Что такое словари в языке Python?
107. Может ли функция `*len()*` быть использована при работе со словарями?
108. Какие методы обхода словарей Вам известны?
109. Какими способами можно получить значения из словаря по ключу?
110. Какими способами можно установить значение в словаре по ключу?
111. Что такое словарь включений?
112. Самостоятельно изучите возможности функции `*zip()*` приведите примеры ее использования.
113. Самостоятельно изучите возможности модуля `*datetime*`. Каким функционалом по работе с датой и временем обладает этот модуль?

Продвинутый уровень

1. Как открыть файл в языке Python только для чтения?
2. Как открыть файл в языке Python только для записи?
3. Как прочитать данные из файла в языке Python?

4. Как записать данные в файл в языке Python?
5. Как закрыть файл в языке Python?
6. Изучите самостоятельно работу конструкции `*with ... as*`. Каково ее назначение в языке Python? Где она может быть использована еще, помимо работы с файлами?
7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?
8. Для чего используется JSON?
9. Какие средства предоставляет язык Python для работы с данными в формате JSON?
10. Каково назначение функций в языке программирования Python?
11. Каково назначение операторов ``def`` и ``return``?
12. Каково назначение локальных и глобальных переменных при написании функций в Python?
13. Как вернуть несколько значений из функции Python?
14. Какие существуют способы передачи значений в функцию?
15. Как задать значение аргументов функции по умолчанию?
16. Каково назначение lambda-выражений в языке Python?
17. Как осуществляется объявление класса в языке Python?
18. Чем атрибуты класса отличаются от атрибутов экземпляра?
19. Каково назначение методов класса?
20. Для чего предназначен метод ``__init__`()` класса?
21. Каково назначение ``self``?
22. Как добавить атрибуты в класс?
23. Как осуществляется управление доступом к методам и атрибутам в языке Python?
24. Что такое наследование и полиморфизм и как они реализованы в языке Python?
25. Каково назначение функции ``isinstance``?

1. Критерии оценивания компетенций

оценка «отлично» выставляется, если студент продемонстрировал высокое умение применять полученные знания на практике через решение конкретной задачи по применению информационных технологий, свободно без затруднений справился с поставленной задачей, показав владение разносторонними приемами и навыками ее выполнения, не допустил ошибок и неточностей;

оценка «хорошо» выставляется, если студент продемонстрировал умение применять полученные знания на практике через решение конкретной задачи по применению информационных технологий, справился с поставленной задачей, показав владение необходимыми приемами и навыками ее выполнения, при этом допустил не более одной ошибки;

оценка «удовлетворительно» выставляется, если студент продемонстрировал посредственное умение применять полученные знания на практике через решение конкретной задачи по применению информационных технологий, с трудом справился с поставленной задачей, при этом допустил не более двух ошибок;

оценка «неудовлетворительно» выставляется, если студент не продемонстрировал умение применять полученные знания на практике через решение конкретной задачи по применению информационных технологий, не справился с поставленной задачей или допустил при ее решении три и более серьезные ошибки.

2. Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным **55**. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0