ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ИМЕНИ. В.Ф. УТКИНА

Кафедра «Вычислительная и прикладная математика»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ
«Алгоритмические языки и программирование»

Направление подготовки

09.03.03 «Прикладная информатика»

Направленность (профиль) подготовки

Прикладная информатика в экономике

Квалификация выпускника – бакалавр

Форма обучения – очная, заочная

Рязань

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов и процедур, предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний обучающихся проводится в форме промежуточной аттестации – зачета в 1-м семестре и курсовой работы и экзамена – во 2-м семестре.

2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;

2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;

3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной

а) описание критериев и шкалы оценивания тестирования:

Шкала оценивания	**Критерий**
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 75 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 60 до 74%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 59%

б) описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	**Критерий**
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя.
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов.
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя.
0 баллов	выставляется студенту, который не смог ответить на вопрос

в) описание критериев и шкалы оценивания практического задания:

Шкала оценивания	**Критерий**
3 балла (эталонный уровень)	Задание решено верно
2 балла (продвинутый уровень)	Задание решено верно, но имеются технические неточности в выполнении
1 балл (пороговый уровень)	Задание решено верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задание не решено

На зачет выносится: тестовое задание, 1 практическое задание и 1 теоретический вопрос.

Студент может набрать максимум 9 баллов.

Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено», «не зачтено».

Оценка «зачтено» выставляется студенту, который набрал в сумме не менее 5 баллов. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий и лабораторных работ.

Оценка «не зачтено» выставляется студенту, который набрал в сумме менее 5 баллов, либо имеет к моменту проведения промежуточной аттестации несданные практические, либо лабораторные работы.

На экзамен выносится: тестовое задание, 1 практическое задание и 1 теоретический вопрос. Студент может набрать максимум 9 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Шкала оценивания	**Критерий**	
отлично (эталонный уровень)	8 – 9 баллов	Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий и лабораторных работ.
хорошо (продвинутый уровень)	6 – 7 баллов	
удовлетворительно (пороговый уровень)	4 – 5 баллов	
неудовлетворительно	0 – 3 баллов	Студент не выполнил всех предусмотренных в течение семестра текущих заданий

Курсовая работа оценивается по принятой в ФГБОУ ВО «РГРТУ» четырехбалльной системе: «неудовлетворительно», «удовлетворительно», «хорошо» и «отлично».

Шкала оценивания	**Критерии оценивания**
«отлично»	*студент должен*: продемонстрировать глубокое усвоение материала; исчерпывающе, последовательно, грамотно и логически стройно изложить теоретический материал; правильно формулировать определения; уметь делать выводы по излагаемому материалу; безупречно ответить на дополнительные вопросы при защите курсовой работы в рамках рабочей программы дисциплины
«хорошо»	*студент должен*: продемонстрировать достаточно полное знание материала; продемонстрировать знание основных теоретических понятий; достаточно последовательно, грамотно и логически стройно изложить материал; уметь сделать достаточно обоснованные выводы; ответить на все вопросы при защите курсовой работы; при этом возможны непринципиальные ошибки;
«удовлетворительно»	*студент должен*: продемонстрировать общее знание материала; знать основную рекомендуемую учебную литературу; уметь строить ответ в соответствии со структурой излагаемого вопроса; показать общее владение понятийным аппаратом дисциплины; уметь устранять допущенные ошибки в ответе на вопросы при защите курсовой работы;
«неудовлетвори-тельно»	*ставится в случае*: незнания значительной части программного материала; не владения понятийным аппаратом; существенных ошибок при изложении учебного материала; неумения строить ответ в соответствии со структурой излагаемого вопроса; неумения делать выводы. Такая оценка ставится студентам, которые не могут продолжить обучение по данной образовательной программе, и если студент нарушил правила защиты курсовой работы (списывал и т.д.).

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Наименование оценочного средства
1 семестр (осенний)		
Знакомство со средой Wing IDE 101	ОПК-2.2 ОПК-7.1 ОПК-7.2	Зачет
Линейные алгоритмы	ОПК-2.2 ОПК-7.1 ОПК-7.2	Зачет
Условный оператор. Многоальтернативное ветвление	ОПК-2.2 ОПК-7.1 ОПК-7.2	Зачет
Цикл с параметром for. Вычисление последовательностей с использованием цикла с заголовком. Конечные суммы и произведения	ОПК-2.2 ОПК-7.1 ОПК-7.2	Зачет
Оператор цикла с предусловием while. Задача табулирования функции	ОПК-2.2 ОПК-7.1 ОПК-7.2	Зачет
Вложенные циклы	ОПК-2.2 ОПК-7.1 ОПК-7.2	Зачет
2 семестр (весенний)		
Списки (одномерные массивы). Кортежи	ОПК-2.2 ОПК-7.1 ОПК-7.2	Экзамен, КР
Множества	ОПК-2.2 ОПК-7.1 ОПК-7.2	Экзамен, КР
Процедуры и функции	ОПК-2.2 ОПК-7.1 ОПК-7.2	Экзамен, КР
Матрицы	ОПК-2.2 ОПК-7.1 ОПК-7.2	Экзамен, КР
Строки	ОПК-2.2 ОПК-7.1 ОПК-7.2	Экзамен, КР
Словари	ОПК-2.2 ОПК-7.1 ОПК-7.2	Экзамен, КР
Файлы. Модули	ОПК-2.2 ОПК-7.1 ОПК-7.2	Экзамен, КР

Для заочной формы обучения дополнительно предусмотрены контрольные работы в 1 и 2 семестре, включающие все контролируемые разделы (темы) дисциплины.

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1 Промежуточная аттестация (зачет и экзамен)

ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности
ОПК-2.2 Использует при решении задач профессиональной деятельности современные информационные технологии и программные средства, в том числе отечественного производства

а) типовые тестовые вопросы закрытого типа

1. Отечественный пакет офисных программ это:
   - Microsoft Office
   - OpenOffice
   - LibreOffice
   - ***МойОфис***
2. Отечественная операционная система это:
   - Xubuntu
   - ***Astra Linux***
   - Linux Mint
   - CentOS
3. На какой операционной системе работает язык Python?
   - Windows
   - Linux
   - MacOS
   - ***На всех перечисленных***
4. Как расшифровывается аббревиатура IDE?
   - ***интегрированная среда разработки***
   - индивидуальный определяющий номер
   - параллельный интерфейс подключения накопителей
5. Что такое компилятор

   это компьютерная программа, которая переводит компьютерный код, написанный на одном языке программирования (исходный язык), на другой язык (целевой язык)

   это компьютерная программа, которая преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд
6. Что такое транслятор

   это компьютерная программа, которая переводит компьютерный код, написанный на одном языке программирования (исходный язык), на другой язык (целевой язык)

   это компьютерная программа, которая преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд
7. Что такое API?
   - Это система вывода текста в командную оболочку.
   - Это ядро операционной системы.
   - ***Это программный интерфейс, с его помощью программы общаются друг с другом.***
   - Это интерфейс подключенного к компьютеру принтера.
8. Что такое лог программы?
   - Файл с инструкцией по ее запуску.
   - ***Файл с информацией о ходе ее работы.***
   - Файл с информацией о текущей версии программы.
   - Файл с информацией об авторе и лицензионных правах на программу.
9. Чем отличается IDE и простой текстовый редактор?
   - IDE – это подвид текстовых редакторов, способных еще и работать с графикой.
   - ***IDE – это программа, включающая в себя не только текстовый редактор, но и другие необходимые для разработки программы.***
   - IDE – это один из видов текстовых редакторов, как Блокнот или Word.
   - IDE – это графический редактор, а не текстовый.

10. Кто разработал язык Python?
    Джеймс Гослинг
    Гвидо ван Россум
    Бьёрн Страуструп
    Никлаус Вирт
11. Каким главным достоинством обладает язык Python?
    простота синтаксиса и команд
    высокая скорость выполнения программ
    статическая типизация данных
12. Язык Python
    компилируемый
    интерпретируемый
13. Какая из перечисленных сред программирования не поддерживает язык Python?
    IDLE
    PyCharm
    Visual Studio Code
    Neovim
    Wing
14. Какие кодировки исходного текста программы поддерживает интерпретатор Python:
    большинство кодировок, распространенных сегодня
    ASCII, Latin-1, UTF-8
    ASCII

б) типовые тестовые вопросы открытого типа

1. Как называется текстовый процессор в МойОфис? (Документы)
2. Язык Python интерпретируемый или компилируемый? (Интерпретируемый)
3. На базе какого ядра построена отечественная ОС? (на ядре Linux)
4. Какие возможности представляет современная среда программирования (текстовый редактор, отладка, компилятор и др.)?
5. Причины и типы ошибок при запуске и выполнении программы (синтаксические ошибки и ошибки на этапе исполнения программы).
6. Что такое справка и контекстная справка? (F1 и Ctrl+F1)
7. Как работает автоматическая подстановка кода (Ctrl+Пробел) в современных средах программирования.
8. Опишите, что такое компилятор и укажите принцип его работы. (Компилятор – это программа, которая преобразует исходный код программы, написанной на одном языке программирования (называемом исходным языком), в эквивалентный код на другом языке (называемом целевым языком), который может быть выполнен на целевой платформе (например, компьютере или микроконтроллере). Основная задача компилятора – это перевести исходный код программы в машинный код или в код на низкоуровневом языке, который может быть понятен компьютеру. Этот процесс включает лексический анализ (разбиение исходного кода на лексемы), синтаксический анализ (построение дерева синтаксического разбора), семантический анализ (проверка правильности использования языковых конструкций) и генерацию кода (создание эквивалентного кода на целевом языке).)
9. Опишите, что такое интерпретатор и укажите принцип его работы. (Интерпретатор – это программа, которая выполняет исходный код программы построчно, без предварительной компиляции. Он читает и анализирует каждую строку исходного кода программы и непосредственно выполняет соответствующие инструкции. В отличие от компилятора, который преобразует весь исходный код программы в машинный код или в код на низкоуровневом языке, интерпретатор выполняет программу "на лету". Он обычно работает с высокоуровневыми языками программирования, такими как Python, JavaScript или Ruby.)

ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ОПК-7.1 Понимает требования к алгоритмам, суть процесса алгоритмизации задач

а) типовые тестовые вопросы закрытого типа

1. Какого вида цикла нет в языке Python?

С предусловием
С постусловием
С заголовком

2. Какого алгоритма сортировок не существует?

пузырьком
вставками
выбором
Вассермана
Шелла.

3. Алгоритм это – …

графическая схема
любая последовательность действий
конечная последовательность действий
конечная последовательность действий, приводящая к однозначному решению задачи

4. К средствам записи алгоритма не относятся:

словесная форма записи
диаграмма Эйлера-Венна
схема алгоритма
структурограмма
псевдокод

5. Что означает символ ромб в схеме алгоритма ?

Процесс
Решение
Модификация

6. При изображении каких конструкций языка программирования используется символ ромб?

Условный оператор
Оператор выбора
Оператор ввода
Оператор вывода
Циклы

7. Верно ли утверждение, что нисходящим проектированием или проектированием «сверху вниз» называется разработка алгоритма с разбиения сложной исходной задачи на отдельные более простые подзадачи, решение которых может быть представлено в общей структуре алгоритма функционально независимыми блоками.

да
нет

8. Что называется разветвляющимися вычислительными процессами?

многальтернативные ветвления
вычислительные процессы, в которых в зависимости от тех или иных условий должны выполняться различные этапы вычислений
алгоритмы с деревьями поиска

9. Можно ли решить задачу поиска корней уравнения с помощью программирования?

да, с помощью итерационных циклов
нет
да, с помощью рекурсии

10. Может ли при реализации вложенных циклов внешний цикл закачиваться раньше, чем внутренний цикл?

нет
да

11. Алгоритм называют циклическим:

если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
если ход его выполнения зависит от истинности тех или иных условий
если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий

12. Определите, какой тип алгоритмической структуры необходимо применить, если последовательность команд выполняется или не выполняется в зависимости от условия:

цикл
ветвление
линейный

13. Алгоритмом можно считать:

описание процесса решения квадратного уравнения
расписание уроков в школе
технический паспорт автомобиля
список класса в журнале

14. Как называется свойство алгоритма, которое означает, что данный алгоритм применим к решению целого класса задач?

понятность
определённость
результативность
массовость

б) типовые тестовые вопросы открытого типа

1. Какой оператор используется для вывода значения на экран в языке Python? (print)
2. Какой оператор используется для чтения значения с клавиатуры значения в языке Python? (input())
3. Что такое линейный алгоритм? (Линейный алгоритм — набор команд (указаний), выполняемых последовательно во времени друг за другом.)
4. Что такое разветвляющийся алгоритм? (алгоритм, содержащий хотя бы одно условие, в результате проверки которого может осуществляться разделение на несколько альтернативных ветвей алгоритма.)
5. Какие операторы языка реализуют алгоритмы с ветвлениями в языке Python? (if, match case, операторы безусловного перехода goto)?
6. Какие бывают виды циклов есть в языке Python?
7. Какие есть примитивные типы данных в в языке Python (числовые int float complex, логические bool…)?
8. Какие есть составные (структурные) типы данных в языке Python (списки, строки, кортежи, файлы, словари)?
9. В каком случае лучше реализовать бинарный файл, а в каком строковый? (если в файле нужно хранить строковую информацию, то текстовый, если не хочется хранить данные в файле в читаемом виде, то можно их записать в бинарный файл)
10. Приведите пример задачи, которую можно решить через итерационную формулу, а можно с помощью рекурсии. (вычисление факториала)
11. Каковы общие принципы организации вложенных циклов? (циклы могут быть любых видов, внешний цикл не должен заканчиваться раньше внутреннего)
12. Напишите программу на языке Python, которая суммирует два целых числа и выводит результат на экран.

```
a = int(input())
b = int(input())
print('Сумма = ', a + b)
```

13. Напишите программу на языке Python, которая вычисляет среднее арифметическое трех чисел и выводит результат на экран.

```
a = float(input())
b = float(input())
c = float(input())
average = (a + b + c) / 3
print('Среднее арифметическое = ', average)
```

14. Составить словестную запись алгоритма алгоритм определения максимального числа из трех z = max(a, b, c) или записать в виде программы на языке Python.

Словесная запись алгоритма:

Начало.
1. Ввести a, b, c.
2. Если a > b, то z := a; иначе z := b.
3. Если c > z, то z := c.
4. Вывести z.

Конец.

15. Составить словестный алгоритм определения остатка от деления двух целых неотрицательных чисел A и B, где $B \neq 0$ или написать программу на языке Python.

Можно реализовать следующий алгоритм.

Начало.

1. Ввести A, B.
2. Если A < B, то перейти к п. 5; иначе перейти к п. 3.
3. A:= A - B.
4. Перейти к п. 2.
5. ОСТ:= A.
6. Вывести ОСТ.

Конец.

Если использовать сущность и структуру циклических алгоритмов, можно записать алгоритм в более компактном виде, например:

Начало.

1. Ввести A, B.
2. Пока $A \geq B$ выполнять A:= A - B.
3. ОСТ:= A.
4. Вывести ОСТ.

Конец.

16. Что такое жадный алгоритм? (Жадный алгоритм – алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.)

17. Что такое рекурсивный алгоритм? (Рекурсивный алгоритм – это алгоритм, который вызывает сам себя.)

ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ОПК-7.2 Выполняет разработку алгоритмического и программного обеспечения для решения прикладных задач

а) типовые тестовые вопросы закрытого типа

1. Что будет в результате выполнения программы

```
a = 5
b = a + 5
a = b * 100
print(a)
```

25

250

2500

1000

2. Что будет в результате выполнения следующего действия (23%2)

1

10

0

3. Что будет в результате выполнения следующего действия (25//3)

8

10

8.3

4. Что будет в результате выполнения программы при a = 10, b = 20?

```
a = int(input())
b = int(input())
if a < b:
```

```
        print(a)
else:
        print(b)
```
10

20

30

-10

5. Что будет выведено в результате выполнения программы при a = 10, b = 20, c=30?

```
a = int(input())
b = int(input())
c = int(input())
if a = = b and b = = c and c = = a:
        print('треугольник равносторонний')
else:
        print('треугольник  не равносторонний')
```
равносторонний

равнобедренный

не равнобедренный

5. Что будет выведено в результате выполнения программы?

```
z = 5
z1 = z**3
print(z1)
```
5

25

125

256

6. Укажите, сколько раз выполнится тело цикла с данным заголовком?

```
i = 1
while i <=10:
        i = i + 1
```
3

4

10

11

12

7. Имеется кортеж вида T = (4, 2, 3). Какая из операций приведёт к тому, что имя T будет ссылаться на кортеж (1, 2, 3)?

```
T[0] = 1
T = (1) + T[1:]
```
T = (1,) + T[1:]

```
T.startswith(1)
```
8. Для чего в Python используется встроенная функция enumerate()?

Для определения количества элементов последовательности

Для одновременного итерирования по самим элементам и их индексам.

Для сортировки элементов по значениям id.

9. Что выведет интерпретатор для следующей программы (версия Python 3.6+)?

```
def get_name_and_decades(name, age):
   print(f"My name is {name} and I'm {age / 10:.5f} decades old.")
get_name_and_decades("Leo", 31)
```

My name is Leo and I'm 31.00000 decades old.

My name is Leo and I'm 3.1 decades old.

Исключение: перед строкой стоит лишняя буква f.

My name is Leo and I'm 3.10000 decades old.

10. Необходимо собрать и вывести все уникальные слова из строки рекламного текста. Какой из перечисленных типов данных Python подходит лучше всего?

кортеж (tuple)

список (list)

множество (set)

словарь (dict)

11. Учёт зверей в зоопарке ведётся с помощью приведённого ниже списка словарей. Какая из строчек кода выведет структуру, отсортированную в порядке увеличения возрастов животных?

```
animals = [
   {'type': 'penguin', 'name': 'Stephanie', 'age': 8},
   {'type': 'elephant', 'name': 'Devon', 'age': 3},
   {'type': 'puma', 'name': 'Moe', 'age': 5},
]
```

sorted(animals, key='age')

Ни один вариант не является верным, два словаря нельзя сравнивать друг с другом.

sorted(animals, key=lambda animal: animal['age'])

sorted(animals)

12. Какой результат выведет следующий код?

```
def f(a, *pargs, **kargs): print(a, pargs, kargs)
f(1, 2, 3, x=4, y=5)
```

1, 2, 3, {'x': 4, 'y': 5}

1 (2, 3) {'x': 4, 'y': 5}

1, 2, 3, 'x=4', 'y=5'

1, 2, 3, 4, 5

б) типовые тестовые вопросы открытого типа

1. Какое ключевое слово используется для объявления и определения новых функций в Python? (def)

2. Как вывести список методов и атрибутов объекта x? (функция dir, она доступна в стандартной библиотеке Python)

3. Как можно более кратко представить следующую запись?

```
if X:
   A = Y
else:
   A = Z
```

(A = Y if X else Z)

4. Почему инструкция a = (i for i in range(n)) , если n = 10**6, выполнится быстрее, чем инструкции

```
a = list(i for i in range(n))
a = [i for i in range(n)]
a = {i for i in range(n)}
```

(В круглых скобках чистое выражение-генератор, оно не загружает в память коллекцию, поэтому присваивание происходит быстрее остальных случаев, ведь сами элементы последовательности не

создаются. В остальных случаях происходит создание коллекция «на месте»)

5. Что выведет на экран следующий код и почему?
```
a, *b, c = [1, 2]
print(a, b, c)
```
(Звёздочка соответствует оператору распаковки последовательности. Два числа присваиваются переменным, а оставшаяся часть последовательности (в данном случае пустой список) присваивается переменной со звёздочкой. Поэтому правильный ответ – 1 [] 2)

6. С помощью Python нужно записать данные в файл, но только в том случае, если файла ещё нет. Какой режим указать в инструкции open()? Что произойдет, если файл уже существует?
('x'. Если файл уже существует, будет вызвано исключение)

7. Для чего в пакетах модулей python в файле __init__.py служит список __all__?
(Список определяет, что экспортировать, когда происходит импорт с помощью from *)

8. Что выведет следующий код и почему?
```
try:
   raise IndexError
except IndexError:
   print('Получено исключение.')
else:
   print('Но в этом нет ничего страшного.')
```
(Генерируемые пользователем исключения перехватываются тем же способом, что и сиключения, которые генерирует интерпретатор Python. Здесь намеренно инициируется встроенное исключение IndexError. Так как в блоке try было вызвано исключение, блок else выполняться не будет. Поэтому будет выведена строка ‘Получено исключение’)

9. Какой оператор используется для присваивания значения переменной в Python? (=)
10. Какой оператор используется для проверки неравенства двух значений в Python? (<> или !=)
11. Что такое алгоритмизация (процесс разработки и описания алгоритма решения какой-либо задачи).
12. Что такое схема алгоритма (графический способ его представления с элементами словесного описания)?
13. Обоснуйте выбор типа цикла для решения задачи табулирования функции на языке Python: цикл с предусловием или цикл с параметром? (первый вид цикла подходит лучше, не требуется введение дополнительной переменной, определяющей число шагов).
14. Обоснуйте, в каком случае программное решение требует использование подпрограмм (функций) в языке Python? (В виде подпрограммы оформляют некоторую часть алгоритма, решающую законченную задачу. У такой задачи должны быть входные данные, которые ей необходимы для вычислений, и выходные данные, которые она вернет в алгоритм, из которого она была вызвана)
15. Что такое итерационные циклы? (это алгоритм, который находит решение за несколько итераций (шагов), на каждом шаге утоняя решение, чтобы оно удовлетворяло заданной погрешности). Приведите примеры решения задач численных методов, реализуемых через итерационные циклы (вычисление корней уравнения, вычисление бесконечных сумм…)
16. Напишите программу, которая выводит таблицу умножения для заданного числа.
```
num = int(input('Введите число:'))
for i in range(1, 11):
        print(num, ' x ', i, ' = ', num * i)
```
17. Напишите программу, которая вычисляет факториал заданного числа.
```
num = int(input('Введите число:'))
factorial = 1
for i in range(1,  num+1):
        factorial = factorial * i
print('Факториал числа ', num, ' = ', factorial)
```

18. Что такое тело цикла? (Это группа команд, повторяющаяся некоторое число раз и расположенная внутри цикла.)

19. Если говорят, что программа “зациклилась”, то это значит… (тело цикла выполняется бесконечно долго)

4.2 Типовые контрольные вопросы и задания к зачету (1-й семестр)

1. Основы алгоритмизации
   - Понятие алгоритма
   - Алгоритмическая система
   - Алгоритмизация
   - Средства записи алгоритмов
   - Технология разработки алгоритмов
   - Структуры алгоритмов
2. Программирование линейных алгоритмов
   - Алфавит Python
   - Стандартные функции Python
   - Функция пользователя
   - Переменные и идентификаторы
   - Понятие типа данных
   - Целые типы
   - Вещественные типы
   - Комплексный тип
   - Строковый тип
   - Логический тип
   - Арифметическое выражение
   - Оператор присваивания
   - Структура программы
   - Стиль программирования
   - Комментарии и документирование программы
3. Программирование алгоритмов разветвляющейся структуры.
   - Логическое выражение
   - Оператор условного перехода
   - Оператор варианта
4. Программирование программ циклической структуры
   - Цикл с известным числом повторений
   - Цикл с предусловием
   - Итерационные циклы
5. Организация ввода-вывода
6. Отладка программ
7. Вложенные циклы

4.3 Типовые вопросы к экзамену по дисциплине (2-й семестр)

1. Одномерные и двумерные массивы. Примеры обработки массивов. Типы данных список и кортеж.
2. Простейшие подпрограммы пользователя. Синтаксис. Пример.
3. Использование подпрограмм для обработки массивов.
4. Строки. Использование подпрограмм для обработки строк
5. Модули.
6. Файлы (текстовые и бинарные).
7. Множества.
8. Словари

4.4 Типовые задачи на зачет и экзамен по дисциплине

1. Протабулировать функцию от заданных значений аргумента

$$Y = \frac{x}{2}\sum_{n=1}^{8}\frac{x^{n+1}}{n+2} + \frac{x+1}{3}\sum_{n=1}^{7}\frac{x^{2n}}{n+3};$$

$$x = 0{,}1(0{,}1)1.$$

2. Протабулировать функцию от заданных значений аргумента

$$Y = \begin{cases} \sum_{n=0}^{50}\frac{1}{(2n+1)^2}, & \text{если} \quad x < 0{,}5; \\ \sum_{n=0}^{10}\ln(x)\cdot\sin k(x-a), & \text{если} \quad x \ge 0{,}5; \end{cases}$$

$$a \quad - \quad \text{задается произвольно};$$

$$x = 0{,}4[0{,}2]1{,}8.$$

3. В одномерном массиве целых чисел подсчитать количество чисел, кратных 5.
4. В одномерном массиве целых чисел подсчитать количество чисел, заканчивающихся на 3.
5. В произвольно заданном одномерном массиве поменять местами максимальное и минимальное числа.
6. В произвольно заданном одномерном массиве подсчитать количество отрицательных чисел и заменить их на средне – арифметическое значение отрицательных чисел
7. В произвольно заданном одномерном массиве заменить отрицательные числа на их абсолютную величину.
8. В произвольно заданном одномерном массиве определить число с минимальной суммой цифр и вывести его адрес.
9. Составить подпрограмму определения минимума функции $y = f(x)$, заданной в дискретных точках отрезка $[a, b]$ с постоянным шагом h. Используя ее, найти минимумы функций $y_1=\sin^2 x\cdot\cos x - \ln|x|$, $y_2=x^2-2x+8$.
10. Составить подпрограмму вычисления определенного интеграла по формуле прямоугольников $\int_a^b f(x)dx \approx \frac{b-a}{n}\sum_{i=0}^{n-1} f(a+i\frac{b-a}{n})$ и, используя ее, вычислить
11. Составить подпрограмму вычисления k-й степени квадратной матрицы из $n \times n$ элементов, используя подпрограмму умножения двух матриц. Найти A^3, B^2, где A, B - произвольные квадратные матрицы.
12. Составить подпрограмму упорядочения элементов одномерного массива по убыванию их значений методом выбора максимального элемента, используя подпрограмму нахождения номера максимального элемента в последовательности чисел. Упорядочить по убыванию массивы X и Y.
13. Определить, является ли вводимая последовательность символов идентификатором.
14. Подсчитать количество сдвоенных символов "сс","нн","лл" во введенном тексте.
15. Разбить произвольный текст на строки определенной длины. При переносе слова предусмотреть вывод дефиса.
16. Дана символьная строка. Посчитать, сколько раз в ней встречается подстрока ABBA.
17. Найти во введенном тексте самое длинное и самое короткое слово.
18. Из заданной строки исключить все символы, входящие в нее более одного раза.

19. Организовать массив записей, содержащий информацию о результатах сдачи последней сессии вашими товарищами. Определить и дополнить данные средним баллом.
20. Организовать массив записей, содержащий информацию о месте жительства нескольких ваших товарищей. Предусмотреть возможность выдачи адреса по введенной фамилии.
21. Сформировать файл, содержащий фамилии нескольких студентов. Добавить к полученному файлу фамилии еще 2-3 студентов.
22. Записать в файл оценки (в баллах), полученные некоторым студентом на экзаменах в течение всех сессий. Добавить в начало файла оценки, полученные на вступительных экзаменах.
23. Записать в файл оценки (в баллах), полученные некоторым студентом на экзаменах в течение всех сессий, и определить средний балл.
24. Сформировать два файла. В один из них поместить фамилии пяти ваших знакомых, а в другой – номера их телефонов. Составить программу, которая по фамилии вашего знакомого определяет номер его телефона.

4.5 Типовые темы курсовых работ

Тема 1. Проектирование обучающе-контролирующих программ.
Тема 2. Информационные системы
Тема 3. Игровые программы
Тема 4. Графические программы
Тема 5. Работа с матрицами. Рекомендуется для студентов заочной формы обучения.

Оператор ЭДО ООО "Компания "Тензор"
ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
СОГЛАСОВАНО **ФГБОУ ВО "РГРТУ", РГРТУ,** Овечкин Геннадий Владимирович, Заведующий кафедрой ВПМ **03.11.26** 22:31 (MSK) Простая подпись