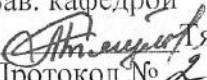


Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»
Рыбницкий филиал

Кафедра информатики и программной инженерии

УТВЕРЖДАЮ

Зав. кафедрой

 Лягульская Л.А., доцент

Протокол № 2 от «13» 09 2021 г.

Фонд оценочных средств

по дисциплине

«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ»

Направление подготовки:

2.09.03.04 «Программная инженерия»

Профиль подготовки:

«Разработка программно-информационных систем»

Квалификация

Бакалавр

Форма обучения

очная

2018 год набора

Разработал:

ст. преподаватель

 А.Б. Глазов

«24» сентябре 2021 г.

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Программирование на языке высокого уровня» у обучающихся должны быть сформированы следующие компетенции:

Код компетенции	Формулировка компетенции
ПК-1	готовность применять основные методы и инструменты разработки программного обеспечения
ПК-3	владение навыками использования различных технологий разработки программного обеспечения

В результате освоения дисциплины студент должен:

Знать:

- формальные методы, технологии и инструменты разработки программного продукта;
- концепции эволюционного развития программного обеспечения;
- концепции и реализации программных процессов.

Уметь:

- устанавливать, тестировать, испытывать и использовать программные средства;
- конструировать программное обеспечение;
- разрабатывать основные программные документы;
- работать с современными системами программирования.

Владеть:

- навыками разработки и отладки программ на языках высокого уровня.

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины и их наименование	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Обзор языков программирования	ПК-1, ПК-3	— текущий — контроль выполнения лабораторных и индивидуальных заданий; — рубежный — контроль выполнения лабораторных и практических работ.
2	Базовые типы данных	ПК-1, ПК-3	
3	Сложные типы данных	ПК-1, ПК-3	
4	Работа с файлами	ПК-1, ПК-3	
5	Обработка классов	ПК-1, ПК-3	
Промежуточная аттестация		Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1		ПК-1, ПК-3	Тест. Вопросы к экзамену

УТВЕРЖДАЮ

зав. кафедрой информатики и программной инженерии,

доцент _____ Л.А. Тягульская

« ____ » _____ 2021 г.

**Вопросы к экзамену
по дисциплине «Программирование на языке высокого уровня»
для студентов IV курса
направления «Программная инженерия»
профиля «Разработка программно-информационных систем»,
7 семестр**

1. Алгоритмы. Свойства алгоритмов. Способы описания алгоритмов. Основные конструкции алгоритмического языка: линейный алгоритм, ветвление, цикл.
2. Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования.
3. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы.
4. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере.
5. Переменные и константы. Объявление объектов данных. Внутреннее представление данных в памяти компьютера.
6. Типы данных. Простые типы данных.
7. Производные типы данных. Структурированные типы данных.
8. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных.
9. Оператор присваивания. Составной оператор. Условный оператор. Оператор выбора.
10. Цикл с постусловием. Цикл с предусловием.
11. Цикл с параметром. Вложенные циклы.
12. Общие сведения о подпрограммах. Определение и вызов подпрограмм.
13. Область видимости и время жизни переменной. Механизм передачи параметров.
14. Рекурсия. Программирование рекурсивных алгоритмов.
15. Основы структурного программирования. Методы структурного программирования.
16. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.
17. Объявление массива. Инициализация. Действия над массивами. Заполнение массива данными. Вывод элементов массива.
18. Удаление и вставка элементов в массив. Обработка массива.
19. Символьный и строковый типы. Объявление типов. Стандартные функции и процедуры для работы со строками.
20. Поиск, удаление, замена и добавление символов в строке. Операции со строками.
21. Понятие множества. Объявление множества. Операции над множествами.
22. Определение типа записи. Правила работы с записями
23. Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа.
24. Файлы произвольного доступа. Порядок работы с файлами произвольного доступа. Создание структуры записи. Открытие и закрытие файла произвольного доступа.
25. Запись и считывание из файла произвольного доступа. Использование файла произвольного доступа.
26. Стандартные процедуры и функции для файлов разного типа.

27. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.
28. Структуры данных на основе указателей.
29. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
30. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства.
31. Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.
32. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.
33. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя.
34. Программирование приложения. Тестирование, отладка приложения. Создание документации.
35. Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события.
36. Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов

Составитель _____ А.Б. Глазов, ст. преподаватель

УТВЕРЖДАЮ

зав. кафедрой информатики и программной инженерии,

доцент _____ Л.А. Тягульская

« ____ » _____ 2021 г.

Образец теста

**для проведения итогового контроля по итогам освоения дисциплины,
а также для контроля самостоятельной работы студента**

Указания: Напишите Вашу фамилию, номер группы и дату. Для ответа на вопрос с выбором варианта ответа достаточно написать номер вопроса и рядом букву, обозначающую правильный вариант из предложенных в тексте ответов на вопрос. Если Вы считаете правильными несколько вариантов ответов, то запишите через запятую соответствующие литеры букв.

1. Какие характеристики можно отнести к языку программирования Python?

- 1) интерпретируемый
- 2) с динамической типизацией
- 3) использующий препроцессор для макроподстановок
- 4) для быстрой разработки приложений

2. Какие характеристики можно отнести к языку программирования Python?

- 1) имеет эффективный компилятор в коды процессора
- 2) использует раннее связывание
- 3) объектно-ориентированный
- 4) универсальный язык программирования

3. Какие характеристики можно отнести к языку программирования Python?

- 1) для быстрой разработки приложений
- 2) богатый и громоздкий синтаксис
- 3) поощряет повторное использование кода

4. Какие характеристики можно отнести к языку программирования Python?

- 1) большая стандартная библиотека модулей
- 2) многоплатформный
- 3) удобен для встраивания в проекты на C/C++

5. Какие из этих утверждений о Python верны?

- 1) программы на Python транслируются в машинные коды, которые затем исполняются
- 2) Python использует промежуточный код
- 3) язык Python применяется для быстрой разработки приложений
- 4) Python имеет обширную библиотеку стандартных модулей

6. Какие парадигмы и стили программирования поддерживает Python?

- 1) логистическое программирование
- 2) структурный стиль
- 3) модульное программирование
- 4) императивное программирование

7. Какие парадигмы и стили программирования поддерживает Python ?

- 1) объектно-ориентированный
- 2) модульный
- 3) программирование в ограничениях
- 4) функциональное программирование

8. Какие парадигмы и стили программирования Python поддерживает?

- 1) объектно-ориентированное программирование
- 2) модульное
- 3) программирование в ограничениях
- 4) функциональное

9. Какие парадигмы и стили программирования Python поддерживает?

- 1) императивное программирование
- 2) функциональное программирование
- 3) объектно-ориентированный подход
- 4) логическое

10. Какие парадигмы Python не поддерживает?

- 1) логическое программирование
- 2) структурный стиль
- 3) модульное программирование
- 4) гибрид парадигм

11. Что будет выведено следующей программой:

```
a = 1
b = 2
a, b = b, a
print a, b, a
```

- 1) 2 1 2
- 2) 1 2 1
- 3) 1 1 1
- 4) 11 11 21

12. Что будет выведено следующей программой:

```
a = "A"
b = "B"
b = b + a
print a + b
```

- 1) сообщение об ошибке в третьей строке
- 2) ABA
- 3) BA
- 4) AB

13. Что будет выведено следующей программой:

```
A = B = 1
a = "A" * 5
b = "B" * 5
a = a + b * 5
print a, b
```

- 1) 30 5
- 2) AAAAAABBBBBBAAAAAABBBBBBAAAAAABBBBBBAAAAAABBBBBB, BBBBB
- 3) AAAAAABBBBBBAAAAAABBBBBBAAAAAABBBBBBAAAAAABBBBBB BBBBB
- 4) AAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBB BBBBB

14. Что будет выведено следующей программой:

```
a = "AB"
b = "BC"
print "%sa, b" % a, b
```

- 1) "AB", "BC" a, b

- 2) ABa, b BC
- 3) ('AB', 'BC')a, b
- 4) (AB, BC)a, b

15. Что будет выведено в следующем примере:

```
>>> a = [1, 2, 3]
>>> a[1] = a[1:2]
>>> print a[1]
```

- 1) [1]
- 2) [2]
- 3) 2
- 4) [1, 2]

16. Что выведет следующая программа

```
S = 0
for i in range(1, 10):
    if i % 2 == 0:
        S = S + i
print S
```

- 1) 0
- 2) 10
- 3) 20
- 4) 30

17. Что выведет следующая программа:

```
S = 0
for i in range(1, 10):
    if i % 2 == 0:
        S = S + i
print S
```

- 1) 10
- 2) 12
- 3) 14
- 4) 20

18. Что выведет следующая программа:

```
S = 0
for i in range(1, 10, 2):
    if i % 2 == 0:
        S = S + i
print S
```

- 1) 0
- 2) 1
- 3) 10
- 4) 20

19. Что выведет следующая программа:

```
S = 0
for i in range(10, 2, -1):
    if i % 2 == 0:
        S = S + i
print S
```

- 1) 0
- 2) 20
- 3) 28

4) 30

20. Что выведет следующая программа:

```
S = 0
for i in range(1, 10):
    for j in range(1, 3):
        if i == j:
            S = S + i + j
            break
```

```
print S
```

- 1) 0
- 2) 2
- 3) 6
- 4) 12

21. Сколько элементов будет содержать список L (то есть, чему будет равно len(L)) после выполнения следующего кода:

```
L = []
L.append([1,2,3])
L.insert(1, "abc")
del L[0][0]
```

- 1) произойдет ошибка
- 2) 1
- 3) 2
- 4) 3

22. Сколько элементов будет содержать список L (то есть, чему будет равно len(L)) после выполнения следующего кода:

```
L = []
L.extend([1,2,3])
L.insert(1, "abc")
del L[0][0]
```

- 1) 2
- 2) 3
- 3) 4
- 4) 4, но произойдет ошибка в 4-й строке

23. Сколько элементов будет содержать словарь D (то есть, чему будет равно len(D)) после выполнения следующего кода:

```
D = {}
D[1], D[2], D[3] = "ABV"
D[0], D[1] = "AB"
```

- 1) 2
- 2) 3
- 3) 3 и произойдет ошибка в 3-й строке
- 4) 4

24. Сколько видимых символов будет напечатано после выполнения следующего кода:

```
S = "\n123\\"
print S
```

- 1) 4
- 2) 5
- 3) 6
- 4) 7

25. Сколько элементов будет содержать список L (то есть, чему будет равно len(L)) после выполнения следующего кода:

```
L = []
L.append([1,2,3])
L = L*2
```

- 1) произойдет ошибка
- 2) 1
- 3) 2
- 4) 6

26. Какого типа значение получится в результате вычисления следующего выражения:

(" ")

- 1) str (строка)
- 2) tuple (кортеж)
- 3) это синтаксическая ошибка
- 4) unicode (Unicode-строка)

27. Какого типа значение получится в результате вычисления следующего выражения:

(' ,)

- 1) str (строка)
- 2) tuple (кортеж)
- 3) это синтаксическая ошибка
- 4) unicode (Unicode-строка)

28. Какого типа значение получится в результате вычисления следующего выражения:

(, " ")

- 1) str (строка)
- 2) tuple (кортеж)
- 3) это синтаксическая ошибка
- 4) unicode (Unicode-строка)

29. Какого типа значение получится в результате вычисления следующего выражения:

[" "]

- 1) str (строка)
- 2) list (список)
- 3) это синтаксическая ошибка
- 4) array (массив)

30. Какого типа значение получится в результате вычисления следующего выражения:

(r'\u0432')

- 1) str (строка)
- 2) unicode (Unicode-строка)
- 3) tuple (кортеж)
- 4) это синтаксическая ошибка

ОТВЕТЫ

1	2,4	8	2	15	2	22	4	29	2
2	3,4	9	4	16	3	23	4	30	1
3	1,3	10	1	17	4	24	1		
4	1,2,3	11	1	18	1	25	3		
5	2,3,4	12	2	19	3	26	1		
6	2,3,4	13	4	20	3	27	2		
7	1	14	2	21	3	28	3		

УТВЕРЖДАЮ

зав. кафедрой информатики и программной инженерии,

доцент _____ Л.А. Тягульская

« ____ » _____ 2021 г.

Задачи для самостоятельного решения

Интерпретируемые языки и языки с динамической типизацией

1. Найдите и выведите все двузначные числа, которые равны удвоенному произведению своих цифр.
2. Квадрат трехзначного числа оканчивается тремя цифрами, равными этому числу. Найдите и выведите все такие числа.
3. Дано натуральное число n . Выведите в порядке возрастания все трехзначные числа, сумма цифр которых равна n .
4. Даны два четырехзначных числа A и B . Выведите все четырехзначные числа на отрезке от A до B , запись которых является палиндромом.
5. Даны два четырехзначных числа A и B . Выведите в порядке возрастания все четырехзначные числа в интервале от A до B , запись которых содержит ровно три одинаковые цифры.

Введение в Python

1. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите, какое наибольшее число подряд идущих элементов этой последовательности равны друг другу.
2. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите наибольшую длину монотонного фрагмента последовательности (то есть такого фрагмента, где все элементы либо больше предыдущего, либо меньше).
3. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите количество строгих локальных максимумов в этой последовательности. (*Элемент последовательности называется локальным максимумом, если он строго больше предыдущего и последующего элемента последовательности. Первый и последний элемент последовательности не являются локальными максимумами.*)
4. Определите наименьшее расстояние между двумя локальными максимумами последовательности натуральных чисел, завершающейся числом 0. Если в последовательности нет двух локальных максимумов, выведите число 0.
5. В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.

Обзор стандартной библиотеки языка Python

1. Дан список чисел, который могут содержать до 100000 чисел каждый. Определите, сколько в нем встречается различных чисел.
2. Даны два списка чисел, которые могут содержать до 100000 чисел каждый. Посчитайте, сколько чисел содержится одновременно как в первом списке, так и во втором.
3. Даны два списка чисел, которые могут содержать до 10000 чисел каждый. Выведите все числа, которые входят как в первый, так и во второй список в порядке возрастания.
4. Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.
5. Во входном файле (вы можете читать данные из файла input.txt) записан текст. Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки. Определите, сколько различных слов содержится в этом тексте.

Объектно-ориентированное программирование в Python

Создать класс с полями, указанными в индивидуальном задании. Реализовать в классе методы: конструктор по умолчанию; функции обработки данных (1 и 2) функцию формирования строки информации об объекте. Создать класс-потомок с полями, указанными в индивидуальном задании. Реализовать в классе-потомке методы: конструктор; функцию обработки данных; функцию формирования строки информации об объекте.

- 1) *Класс 1:* Дата (три числа): день, месяц, год

Определить, является ли год високосным (кратным 4)

Увеличить дату на 5 дней

Класс 2: Человек: ФИО, телефон, дата рождения,
Количество дней до дня очередного рождения

2) **Класс 1:** Дата (три числа): день, месяц, год

Увеличить год на 1

Уменьшить дату на 2 дня

Класс 2: Работник: ФИО, дата поступления на предприятие

Количество лет работы на предприятии

3) **Класс 1:** Дата (три числа): день, месяц, год

Определить, совпадают ли номер месяца и число дня

Увеличить дату на один месяц

Класс 2: Лекарство: наименование, дата выпуска, фирма

Сколько прошло дней от изготовления лекарства

4) **Класс 1:** Время (три числа): часы, минуты, секунды

Вычислить количество секунд в указанном времени

Увеличить время на 5 секунд

Класс 2: Данные по движению поезда: номер поезда, направление, время отправления

Количество минут до отправления поезда с указанным номером и введенное время

5) **Класс 1:** Время (три числа): часы, минуты, секунды

Вычислить количество полных минут в указанном времени

Уменьшить время на 10 минут

Класс 2: Абонент мобильной связи: фамилия, оператор, текущее время

Определить, является ли время льготным для абонента (время от 0 до 8 часов)

Функциональное программирование в Python

1. Дан файл, содержащий текст на русском языке и некоторые два слова. Определить, сколько раз они встречаются в тексте и сколько из них — непосредственно друг за другом.
2. Даны файл, содержащий текст на русском языке, и некоторые буквы. Найти слово, содержащее наибольшее количество указанных букв.
3. Даны файл, содержащий текст на русском языке, и некоторая буква. Подсчитать, сколько слов начинается с указанной буквы.
4. Дан файл, содержащий текст на русском языке. Найти слово, встречающееся в каждом предложении, или сообщить, что такого слова нет.
5. Дан файл, содержащий зашифрованный русский текст. Каждая буква заменяется на следующую за ней (буква я заменяется на а). Получить в новом файле расшифровку данного текста.