

Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»
Физико-технический институт
Физико-математический факультет
Кафедра высшей и прикладной математики и информатики

УТВЕРЖДАЮ
Заведующий кафедрой-разработчиком

 к.ф.-м.н., доцент Коровай А.В.,
протокол №1 «30» августа 2024 г.

Фонд оценочных средств

по дисциплине

Б1.В.ДВ.02.01 «Современные технологии программирования»

на 2024/2025 учебный год

Направление

01.03.04 «Прикладная математика»

Профиль

Математические и компьютерные методы для современных цифровых технологий

Квалификация (степень) выпускника

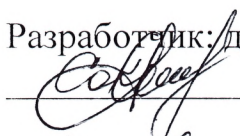
бакалавр

Форма обучения

Очная

ГОД НАБОРА 2022

Разработчик: доцент

 Е.В. Сокольская

«29» августа 2024 г.

Тирасполь 2024

Паспорт фонда оценочных средств по учебной дисциплине

«Современные технологии программирования»

1. В результате изучения дисциплины «Современные технологии программирования» у обучающихся должны быть сформированы следующие компетенции:

Категория (группа) компетенций	Код и наименование	Код и наименование индикатора достижения универсальной компетенции
Обязательные профессиональные компетенции и индикаторы их достижения		
	ПК-4 Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе с учетом возможностей современных информационных технологий и программирования.	ИД-1 _{ПК-4} Знает математические методы, основы программирования и современные компьютерные технологии.
		ИД-2 _{ПК-4} Умеет использовать математический аппарат, основы программирования и современные компьютерные технологии для решения практических задач
		ИД-3 _{ПК-4} Владеет навыками использования математического аппарата, основ программирования и современных компьютерных технологий для решения практических задач профессиональной деятельности.
	ПК-6 Способен разрабатывать алгоритмы, проектировать и разрабатывать компоненты программного обеспечения на основе современных парадигм, технологий и языков программирования.	ИД-1 _{ПК-6} Знает основные языки программирования и основы работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.
		ИД-2 _{ПК-6} Умеет применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.
		ИД-3 _{ПК-6} Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули (темы) и наименование	Код контролируемой компетенции	Наименование оценочного средства
1	Тестирование по Разделу 1 «Основные понятия технологии программирования», по Разделу 2 «Стратегии разработки программных средств и реализующие их модели жизненного цикла», по Разделу 3 «Тестирование программного средства»	ПК-4, ПК-6	Комплект тестовых заданий №1
2	Тестирование по Разделу 4 «Основы программирования на языке Python»	ПК-4, ПК-6	Комплект тестовых заданий №2
3	Тестирование по Разделу 5 «Функциональное программирование»	ПК-4, ПК-6	Комплект тестовых заданий №3

4	Тестирование по Разделу 6 «Объектно-ориентированное программирование»	ПК-4, ПК-6	Комплект тестовых заданий №4
Промежуточная аттестация		Код контролируемой компетенции	Наименование оценочного средства
зачет с оценкой		ПК-4, ПК-6	Список вопросов к зачету с оценкой

Наименование оценочного средства

Комплект тестовых заданий № 1

по Разделу 1. «Основные понятия технологии программирования»

по Разделу 2. «Стратегии разработки программных средств и реализующие их модели жизненного цикла»

по Разделу 3. «Тестирование программного средства»»

по дисциплине «Современные технологии программирования»

1. *Надежность (reliability) ПС - это*

- 1) его способность безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью
- 2) его способность безотказно выполнять определенные функции при заданных условиях
- 3) его способность решать поставленную задачу

2. *Надежность ПС подразумевает ...*

- 1) отсутствие каких-либо ошибок в ПС при его практическом применении
- 2) чтобы ошибки при практическом применении этого ПС в заданных условиях проявлялись достаточно редко
- 3) комфортную работу пользователя с ПС

3. *Альтернативой правильного ПС является ...*

- 1) надежное ПС
- 2) качественное ПС
- 3) коммуникативное ПС

4. *При помощи чего можно убедиться в надежности ПС?*

- 1) тестирования
- 2) практического применения
- 3) отладки

5. *Надежность работы ПС определяется*

- 1) независимостью от технического комплекса системы обработки данных
- 2) независимостью от операционной среды, сетевой технологии обработки данных, специфики предметной области
- 3) корректностью выполнения функций задачи
- 4) бесспорностью и устойчивостью в работе программ, точностью выполнения предписанных функций обработки возможностью диагностики возникающих в процессе работы программ ошибок
- 5) корректностью выхода из программы

6. *Какие интеллектуальные возможности человека, используемые при разработке ПС, выделяет Дейкстра?*

- 1) способность алгоритмически мыслить
- 2) способность к творчеству
- 3) способность к перебору
- 4) способность к абстракции
- 5) способность к математической индукции

7. *n – число элементов системы, если $n < X$, то система будет считаться малой. Чему равно X ?*

8. *Пользовательская документация ПС предназначена для ...*

- 1) разработчиков ПС
- 2) ординарных пользователей ПС
- 3) людей, участвующих в модификации ПС

4) администраторов ПС

9. *Документация по сопровождению ПС предназначена для ...*

- 1) ординарных пользователей ПС
- 2) администраторов ПС
- 3) людей, участвующих в модификации ПС

10. *Отладка ПС - это ...*

- 1) деятельность, направленная на обнаружение ошибок в ПС
- 2) деятельность, направленная на обнаружение и исправление ошибок в ПС
- 3) деятельность, направленная на редактирование ошибок в ПС

11. *Тестирование ПС - это ...*

- 1) процесс выполнения его программ на некотором наборе данных, для которого заранее известен результат применения
- 2) процесс обнаружения и исправления ошибок в ПС
- 3) процесс редактирования ошибок в ПС

12. *Оптимальная стратегия проектирования тестов предполагает разработку ...*

- 1) большего количества тестов по отношению к текстам программ по сравнению с тестами по отношению к спецификациям
- 2) все зависит от целей тестирования
- 3) большего количества тестов по отношению к спецификациям по сравнению с тестами по отношению к текстам программ

13. *Оптимальная стратегия проектирования тестов предполагает разработку ...*

- 1) меньшего количества тестов «черного ящика» по сравнению с тестами «белого ящика»
- 2) одинокого количества тестов «белого ящика» и «черного ящика»
- 3) все зависит от целей тестирования
- 4) меньшего количества тестов «белого ящика» по сравнению с тестами «черного ящика»

14. *Какого вида ошибок не существует?*

- 1) синтаксические
- 2) ошибки времени выполнения
- 3) логические
- 4) структурированные

15. *Синтаксические ошибки, это ошибки*

- 1) ошибки некорректного ввода
- 2) ошибки при выполнении
- 3) ошибки при подсчете

16. *Логические ошибки, это ошибки*

- 1) ошибки вовремя не выполнения
- 2) ошибки при выполнении
- 3) ошибки при подсчете из-за не правильного алгоритма
- 4) ошибки некорректного ввода

17. *Отладка программы - это*

- 1) работы, связанные с обслуживанием программы в процессе ее эксплуатации
- 2) процесс поиска и устранения ошибок в программе, производимый по результатам ее прогона на ПК
- 3) контроль логически сложных участков программы
- 4) определение функциональных возможностей программы

18. К методам «белого ящика» относятся

- 1) метод покрытия операторов
- 2) метод покрытия решений
- 3) метод покрытия условий
- 4) метод комбинаторного покрытия условий
- 5) метод эквивалентных разбиений
- 6) метод анализа граничных значений
- 7) метод предположения об ошибке

19. К методам «черного ящика» (методам функционального тестирования) относятся

- 1) метод эквивалентных разбиений
- 2) метод покрытия операторов
- 3) метод покрытия решений
- 4) метод анализа граничных значений
- 5) метод покрытия условий
- 6) метод комбинаторного покрытия условий
- 7) метод предположения об ошибке

20. Жизненный цикл ПО - это

- 1) время выполнения программного обеспечения
- 2) время создания программного обеспечения
- 3) время работоспособности программного обеспечения
- 4) время эксплуатации программного продукта
- 5) непрерывный процесс, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент полного изъятия его из эксплуатации

Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов - за хотя бы один неверный ответ. Затем первичные баллы (максимум - 10 баллов) пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Наименование оценочного средства

Комплект тестовых заданий №2 по Разделу 4. «Основы программирования на языке Python»

по дисциплине «Современные технологии программирования»

1. *Какие парадигмы и стили программирования поддерживает Python?*

- 1) логическое программирование
- 2) императивное программирование
- 3) структурное программирование
- 4) модульное программирование

2. *Какие из этих утверждений о Python верны?*

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

3. *Что будет выведено на экран после выполнения программы?*

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

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

4. *Что будет выведено на экран после выполнения программы?*

```
a = "AB"  
b = "BC"  
print ("{} a, {} b".format( a, b))
```

- 1) ("AB", "BC")a, b
- 2) «AB», «BC»a, b
- 3) (AB, BC)a. b
- 4) AB a, BC b

5. *Чему будет равно значение переменной c после выполнения этой программы:*

```
a = 20  
b = 3  
b = a // b + b  
c = a % b + a
```

6. *Определите значение переменной a после выполнения фрагмента программы:*

```
a = 10  
if a < 5:  
    a = 5
```

7. *Определите значение переменной a после выполнения фрагмента программы:*

```
a = 10  
if a < 5:  
    a += 12  
else:  
    a -= 7
```

8. Какое число нужно написать вместо многоточия, чтобы цикл выполнялся ровно 3 раза?

```
i = 2
while i <= ...:
    print ( "Привет!" )
    i += 1
```

9. Чему будет равно значение целой переменной *a* после выполнения этого фрагмента программы?

```
a = 10;
for i in range(2):
    a - = 1
```

10. Что будет выведено на экран после выполнения программы?

```
s = "1"
s = s + "0" + s;
s = s + s;
print ( s )
```

11. Как выглядит инструкция добавления элемента *x* в множество *s*?

- 1) s+x.add()
- 2) (s+x).add
- 3) s.add(x)
- 4) x.add(s)

12. Как называется встроенный в языке Python тип данных неупорядоченной коллекции из нуля или более пар ключ-значение?

- 1) dict
- 2) set
- 3) list

13. Неупорядоченными типами данных в Python являются:

- 1) dict
- 2) set
- 3) list
- 4) str

14. Пустой список создается при помощи:

- 1) []
- 2) {}
- 3) ()

15. Дан массив *c* = array([[1,2], [2,3], [4,5]]). Чему равен *c*[1]?

- 1) array([1, 2])
- 2) array([2, 3, 5])
- 3) array([2, 3])
- 4) array([1, 2, 4])

Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов – за хотя бы один неверный ответ. Затем первичные баллы (максимум 10 баллов) пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Наименование оценочного средства

Комплект тестовых заданий №3 по Разделу 5. «Функциональное программирование»

по дисциплине «Современные технологии программирования»

1. *Как определить функцию в Python?*

- 1) с помощью оператора def
- 2) заданием списка строк исходного кода
- 3) с помощью lambda-выражения
- 4) с помощью оператора import

2. *Зачем используют оператор return?*

- 1) чтобы приостановить работу программы
- 2) чтобы организовать цикл
- 3) чтобы перейти по произвольному адресу в программе
- 4) чтобы вернуть произвольное значение, указанное в качестве операнда return, в вызывающую функцию

3. *Укажите правильный вариант функции, возвращающей сумму двух вещественных чисел:*

- 1)

```
def sum(j, k)
    return (j+k)
```
- 2)

```
sum(j, k)
    return (j+k)
```
- 3)

```
sum( )
    j = 0
    k = 0
    sum = j+k
```
- 4)

```
int sum( )
    sum = j+k
```

4. *Как передавать функции информацию?*

- 1) циклом for
- 2) информацию функции передать нельзя!
- 3) при помощи аргументов

5. *Может ли функция содержать более одного оператора return?*

- 1) функция может содержать более одного оператора return
- 2) функция не может содержать более одного оператора return

6. *В чем разница между фактическими и формальными параметрами?*

- 1) формальные параметры могут использоваться только вне тела функции, а фактические – используются как вне функции, так и внутри ее
- 2) формальные параметры определены в заголовке функции, а фактические – значения, с которыми функция вызывается
- 3) нет различий

7. *Укажите правильные объявления функции с двумя аргументами:*

- 1)

```
def sum(a,b # тело функции
```
- 2)

```
def sum(a, b) # тело функции
```
- 3)

```
def sum(a = 0, b = 0) # тело функции
```

8. *Где необходимо описывать локальные переменные функции?*

- 1) в вызывающей программе
- 2) после оператора return
- 3) после имени функции

9. *Может ли функция не иметь аргументов?*

- 1) да
- 2) нет
- 3) только если функция возвращает значение

10. *Сколько функций может быть в программе Python?*

- 1) ни одной
- 2) минимум одна
- 3) не больше 100

11. *Может ли функция возвращать несколько значений:*

- 1) нет
- 2) да

12. *Дан фрагмент кода*

```
a = 5
def gg():
    a = 6
    print (a)
gg()
print (a)
```

В результате работы программы будут выведены числа:

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

Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов – за хотя бы один неверный ответ. Затем первичные баллы (максимум 10 баллов) пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Наименование оценочного средства

Комплект тестовых заданий №4 по Разделу 6. «Объектно-ориентированное программирование»

по дисциплине «Современные технологии программирования»

1. *Соккрытие информации о внутреннем устройстве объекта, при котором вся работа с объектом ведется только через общедоступный интерфейс называется*

- 1) агрегацией
- 2) абстракцией
- 3) декомпозицией
- 4) инкапсуляцией

2. *Корректно ли следующее объявление класса*

```
class myClass:  
    pass
```

- 1) да
- 2) нет

3. *Что напечатает следующий код:*

```
def f(self, x):  
    return x*2  
class C:  
    ff = f  
obj = C()  
print(obj.ff(1))
```

- 1) 0
- 2) 2
- 3) ничего не напечатает
- 4) возникнет ошибка, так как метод класса не может быть объявлен вне класса
- 5) возникнет ошибка, так как метод ff принимает 2 параметра

4. *Как называется отношение, которое имеют следующие два класса:*

```
class A(object):  
    def __init__(self, x):  
        self._mydata = x  
    def m1(self):  
        raise NotImplementedError
```

```
class B(A):  
    def __init__(self, x):  
        super(B, self).__init__(x)  
    def m1(self):  
        return self._mydata
```

- 1) агрегация. Экземпляры A содержат экземпляры класса B
- 2) ассоциация. Экземпляры A содержат ссылки на экземпляры класса B
- 3) наследование. A получается наследованием B
- 4) наследование. B получается наследованием A

5. *Сколько общедоступных методов будет иметь экземпляр класса ABC и что возвратит вызов метода a()?*

```
class A(object):  
    def a(self): return 'a'  
class B(object):  
    def b(self): return 'b'  
class C(object):
```

```

def c(self): return 'c'
class AB(A, B):
    def a(self): return 'ab'
class BC(B, C):
    def a(self): return 'bc'
class ABC(AB, B, C):
    def a(self): return 'abc'

```

- 1) 3, возвратит a
- 2) 3, возвратит ab
- 3) 3, возвратит abc
- 4) 5, возвратит a

6. Для чего нужны классы?

- 1) для определения новых типов в программе
- 2) для упрощения работы со сложными структурами данных
- 3) для упрощения работы с константами
- 4) для соединения данных и операций над ними

7. Что напечатает следующий код:

```

class A:
    def __init__(self, v1, v2):
        self.field1 = v1
        self.field2 = v2
    def __str__(self):
        return self.field1 + " " + self.field2
a = A(3, 4)
print(a)

```

- 1) 3 4
- 2) 4 3
- 3) ошибка

8. Что будет выведено на экран в результате работы следующего кода:

```

class Cars:
    def nothing(self, word):
        print ("Привет " + word)
class BMW (Cars):
    def nothing(self, word):
        print (word + "!")
a = BMW()
a.nothing("Вася")

```

- 1) Вася!
- 2) Привет Вася
- 3) Привет Вася!
- 4) Привет!

9. Команда `root = Tk()` позволяет ...

- 1) создать графическое окно
- 2) создать элемент управления «кнопка»
- 3) закончить работу приложения
- 4) сделать элемент управления с именем `root` активным

10. При помощи какой команды можно создать поле для ввода текста

- 1) `str = Label(text="Введите:")`
- 2) `str = Text(textvariable=message)`
- 3) `str = Edit(textvariable=message)`
- 4) `str = Entry(textvariable=message)`

Критерии оценки:

За каждое правильно выполненное задание назначается 1 балл. В случае задания с выбором нескольких правильных ответов 1 балл дается за полностью выполненное задание, 0 баллов – за хотя бы один неверный ответ. Затем первичные баллы пропорционально переводятся в тестовые баллы согласно технологической карте дисциплины.

Наименование оценочного средства

Список вопросов к зачету с оценкой

по дисциплине «Современные технологии программирования»

1. История языка программирования Python. Компиляция и интерпретация.
2. Лексические и синтаксические соглашения языка программирования Python.
3. Понятие переменной. Типы данных языка программирования Python. Особенности использования, возможные операции.
4. Основные стандартные модули в Python. Импортирование модулей.
5. Условный оператор. Инструкция *if*. Множественное ветвление.
6. Цикл *while* языка программирования Python.
7. Цикл *for* в языке программирования Python.
8. Использование команд *break*, *continue*.
9. Последовательности: строки.
10. Последовательности: списки.
11. Организация работы с вложенными списками.
12. Структуры данных: словари.
13. Структуры данных: кортежи.
14. Структуры данных: множества.
15. Функции в языке программирования Python. Области видимости и вложенные функции. Передача аргументов.
16. Функциональные возможности, присущие Python. Анонимные функции: *lambda*. Отображения функций на последовательности: *map*.
17. Основы программирования модулей. Создание модуля. Использование модулей. Пространство имен модулей.
18. Инструкция *class*. Конструктор класса. Экземпляры класса.
19. Перегрузка операторов.
20. Наследование. Полиморфизм.
21. Обработка исключений. Инструкция *try/except/else*. Инструкция *try/finally*. Исключения на основе строк. Исключения на основе классов. Вложенные обработчики исключений.
22. Технология программирования и информатизация общества. Программа как формализованное описание процесса обработки данных. Понятие программного средства.
23. Надежность программного средства. Технология программирования как технология разработки надежных программных средств.
24. Источники ошибок в программном средстве. Неправильный перевод информации как причина ошибок в программном средстве. Модель перевода.
25. Специфика разработки программных средств. Жизненный цикл программного средства.
26. Понятие внешнего описания, его назначение и роль в обеспечении качества программного средства. Определение требований к программному средству.
27. Документация, создаваемая и используемая в процессе разработки программных средств.
28. Основные правила отладки программного средства. Стратегия проектирования тестов.
29. Тестирование программ методами “белого ящика”. Метод покрытия операторов. Метод покрытия решений. Метод покрытия условий. Метод комбинаторного покрытия условий.
30. Тестирование программ методами “черного ящика”. Метод эквивалентных разбиений. Метод предположения об ошибке. Метод анализа граничных значений.

Практические задания:

1. Вычислить сумму ряда $\sum_{k=5}^{10} \left(\frac{1}{k^2} + k \right)$.
2. Найти цифровой корень целого числа N. Цифровой корень находится суммой через сумму цифр числа до тех пор, пока эта сумма не станет цифрой. Например: цифровой корень числа 34697 равняется 2, т.к. $3+4+6+9+7=29$, $2+9=11$, $1+1=2$.
3. Электронные часы показывают время в формате от 00:00 до 23:59. Подсчитать сколько раз за сутки случается так, что слева от двоеточия показывается симметричная комбинация для той, что справа от двоеточия (например, 02:20, 11:11 или 15:51).
4. Напишите программу, которая вводит строку и выводит ее, сокращая каждый раз на 1 символ до тех пор, пока в строке не останется 1 символ.

5. Дан список целых чисел, состоящий из N элементов. Заполнить его с клавиатуры. Найти сумму отрицательных элементов, модуль которых не превосходит 5.
6. Дан список целых чисел, состоящий из N элементов. Заполнить его с клавиатуры. Найти номер последней пары соседних элементов с разными знаками.
7. Дан список числовых значений, насчитывающий N элементов. Подсчитать количество чисел, делящихся на 3 нацело, и среднее арифметическое чисел с чётными значениями. Поставить полученные величины на первое и последнее места в массиве (увеличив массив на 2 элемента)
8. Имеется список, содержащее натуральные числа из некоторого диапазона. Сформировать два множества, первые из которых содержит все простые числа из данного множества, а второе – все составные.
9. Дан текст, за которым следует точка. В алфавитном порядке напечатать все строчные русские гласные буквы (*а, е, и, о, у, ы, э, ю, я*), входящие в этот текст.
10. Создайте словарь, связав его с переменной school, и наполните его данными, которые бы отражали количество учащихся в десяти разных классах (например, 1а, 1б, 2б, 6а, 7в и т.д.). Реализуйте возможность узнать сколько человек в каком-нибудь классе.

Критерии оценки:

-оценка «**отлично**» выставляется, если все вопросы билета раскрыты в полном объеме с четкими определениями и понятными примерами, полностью решена задача.

-оценка «**хорошо**» выставляется, если из двух вопросов билета один раскрыт в полном объеме с четкими определениями и понятными примерами, второй раскрыт недостаточно полно, задача полностью решена.

-оценка «**удовлетворительно**» выставляется, если из двух вопросов билета один раскрыт в полном объеме с четкими определениями и понятными примерами, второй раскрыт недостаточно полно, задача не решена.

-оценка «**неудовлетворительно**» выставляется, если не было дано ответов на вопросы и задача не решена.