Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Кислова Наталья Николаевна
Должность: Проректор по УМР и качеству образования
Дата подписания: 06.06.2025 12:29:57
Уникальный программный ключ:
52802513f5b14a975b3e9b13008093d5726b159bf6064f865ae65b96a966c035

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Самарский государственный социально-педагогический университет»
Информатики, прикладной математики и методики их преподавания

Утверждаю
Проректор по учебно-методической
работе и качеству образования
_____ Н.Н. Кислова

Пугач Ольга Исааковна

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения промежуточной аттестации по дисциплине
«Основы программирования»

Направление подготовки:
44.03.01 Педагогическое образование
Направленность (профиль):
«Информатика»

Квалификация выпускника
Бакалавр

Рассмотрено
Протокол №10 от 27.05.2025
Заседания кафедры информатики, прикладной математики и методики их преподавания

Одобрено
Начальник Управления образовательных программ
Н.А. Доманина

Самара 2025

Пояснительная записка

Фонд оценочных средств (далее – ФОС) для промежуточной аттестации по дисциплине «Основы программирования» разработан в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению подготовки 44.03.01 Педагогическое образование. Утвержден приказом Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 121 основной профессиональной образовательной программой высшего образования «Информатика» с учетом требований профессионального стандарта 01.001 «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)», утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 октября 2013 г. № 544н (зарегистрирован Министерством юстиции Российской Федерации 6 декабря 2013 г., регистрационный № 30550), с изменениями, внесенными приказами Министерства труда и социальной защиты Российской Федерации от 25 декабря 2014 г. № 1115н (зарегистрирован Министерством юстиции Российской Федерации 19 февраля 2015 г., регистрационный № 36091) и от 5 августа 2016 г. № 422н (зарегистрирован Министерством юстиции Российской Федерации 23 августа 2016 г., регистрационный № 43326), 01.003 «Педагог дополнительного образования детей и взрослых» утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 22 сентября 2012 г. № 652н от 22.09.2021 г. (Зарегистрировано в Минюсте России 17.12.2021 N 66403).

Цель ФОС для промежуточной аттестации – установление уровня сформированности компетенций УК-1.

Задачи ФОС для промежуточной аттестации - контроль качества и уровня достижения результатов обучения по формируемым в соответствии с учебным планом компетенциям:

УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

УК-1.1. Анализирует задачу, выделяя этапы ее решения, действия по решению задачи

Знает: этапы решения задачи на компьютере.

Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи.

УК-1.2. Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи

Знает: способы формализации алгоритмов на языках программирования высокого уровня.

Умеет: осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи.

УК-1.3. Рассматривает различные варианты решения задачи, оценивает их преимущества и риски

Знает: о парадигмах программирования; основные конструкции языков программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.

Умеет: строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня.

УК-1.4. Грамотно, логично, аргументированно формирует собственные суждения и оценки; отличает факты от мнений, интерпретаций, оценок в рассуждениях других участников деятельности

Знает: основные виды ошибок, возникающих при решении задачи.

Умеет: комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи.

УК-1.5. Определяет и оценивает практические последствия возможных вариантов решения задачи

Умеет: выполнять оценку сложности алгоритмов, проводить анализ и оценивание полученных результатов.

Требование к процедуре оценки:

Требование к процедуре оценки:

Помещение: особых требований нет

Расходные материалы: лист бумаги, ручка

Доступ к дополнительным справочным материалам: не предусмотрен

Нормы времени: 130 мин

Проверяемый (ые) результат (ы) обучения:

УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

Проверяемый индикатор достижения компетенции:

Проверяемый (ые) результат (ы) обучения:

УК-1.1. Анализирует задачу, выделяя этапы ее решения, действия по решению задачи

Знает: этапы решения задачи на компьютере.

Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи.

Проверяемый индикатор достижения компетенции:

УК-1.2. Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи

Проверяемый (ые) результат (ы) обучения:

Знает: способы формализации алгоритмов на языках программирования высокого уровня.

Умеет: осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи.

Проверяемый индикатор достижения компетенции:

УК-1.3. Рассматривает различные варианты решения задачи, оценивает их преимущества и риски

Проверяемый (ые) результат (ы) обучения:

Знает: методы процедурного, объектно-ориентированного, функционального и визуального программирования; основные конструкции языков программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.

Умеет: строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня.

Проверяемый индикатор достижения компетенции:

Проверяемый (ые) результат (ы) обучения:

УК-1.4. Грамотно, логично, аргументированно формирует собственные суждения и оценки; отличает факты от мнений, интерпретаций, оценок в рассуждениях других участников деятельности

Знает: основные виды ошибок, возникающих при решении задачи.

Умеет: комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.

Проверяемый индикатор достижения компетенции:

Проверяемый (ые) результат (ы) обучения:

УК-1.5. Определяет и оценивает практические последствия возможных вариантов решения задачи

Умеет: выполнять оценку сложности алгоритмов, проводить анализ и оценивание полученных результатов.

Тип (форма) задания:

Пример типовых заданий (оценочные материалы):

Содержание задания:

A1. Какие из нижеперечисленных парадигм программирования поддерживает язык Python
a) объектно-ориентированное;
b) структурное;
c) функциональное;
d) все вышеперечисленные?

A2. Язык Python является:
a) регистрозависимым;
b) компилируемым;
c) высокоуровневым;
d) всё вышеперечисленное.

A3. Инструкция list задает
a) кортеж;
b) список;
c) множество;
d) строку.

A4. Инструкция tuple задает
a) кортеж;
b) список;
c) множество;
d) строку.

A5. Равенство в Python обозначается конструкцией
a) <>
b) = =
c) !=
d) =!

A6. Что будет является результатом выполнения следующих строк кода
```
x = input ('введите число')
print(type(x))
```
a) <class 'int'>;
b) <class 'float'>;

c) <class 'str'>;
d) класс переменной будет зависеть от введенных данных?

A7. Какая библиотека используется для работы с регулярными выражениями в Python?
a) request;
b) requests;
c) re;
d) random?

A8. С помощью какой инструкции в языке Python определяются функции
a) function;
b) procedure;
c) def;
d) proc?

A9. Функция open ('text.txt', 'r') открывает файл text.txt…
a) На запись;
b) На чтение;
c) На дозапись;
d) На чтение и запись.

A10. Что будет является результатом выполнения следующих строк кода
```
b=6
a=b=8
print(a)
```
a) 6;
b) 8;
c) False;
d) True?

A11. Чему будет равна переменная sum результате выполнения следующего участка кода
```
 sum = 0
for i in range(1, n + 1):
    sum += i
```
a) n!;
b) (n+1)!;
c) n;
d) n+1?

A12. S=’программа’. Результатом выполнения команды list(S) будет
a) ['п', 'р', 'о', 'г', 'р', 'а', 'м',];
b) ['п', 'р', 'о', 'г', 'р', 'а', 'м', 'м', 'а'];
c) [‘программа’];
d) возникнет ошибка.

A13. S=’5’. Каким будет результат выполнения команды S*3
a) 15;
b) ‘15’;
c) ‘555’;
d) возникнет ошибка?

A14. Какая из следующих конструкций вернёт длину слова s
a) s.length;
b) s.length();
c) s.len;
d) len(s)?

A15. S=’программа’. Каким будет результат выполнения команды S[::-1]
a) ['п', 'р', 'о', 'г', 'р', 'а', 'м', 'м', 'а']
b) 'аммаргорп'
c) ‘программ’
d) ‘рограмма’?

A16. Что будет выведено на экран в результате выполнения следующего участка кода

```
a=6.7
b = a // 2
pt=rint(a)
```

a) 3;
b) 3.35;
c) Syntax Error;
d) 0.7?

A17. Что будет выведено на экран в результате выполнения следующего участка кода

```
 a = [1,2, None,(),[],]
print(len(a))
```

a) 3;
b) 5;
c) Syntax Error;
d) Infinity?

A18. Что будет выведено на экран в результате выполнения следующего участка кода

```
x=[1,2,3,4,5,6]
Print(x[3])
```

a) 3;
b) 4;
c) Syntax Error;
d) [1,2,3]?

A19. Что будет выведено на экран в результате выполнения следующего участка кода

```
x = [1, 2, 7]
y = x
y[-1] = ['a', 'b']
print x
```

a) [1, 2, 'a', 'b'];
b) [1, 2, ['a', 'b']];
c) ['a', 'b', 1, 2];
d) [[1, 2], 'a', 'b']?

A20. Что будет выведено на экран в результате выполнения следующего участка кода?

```
a=6
while a<10:
    print(str(a)*3,end=’’)
    a+=2
```

a) 1824;
b) 18 24;
c) 666888;
d) 666 888.

Оценочный лист к типовому заданию А

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
d	d	b	a	b	d	c	c	b	b
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
a	b	c	d	b	a	b	b	b	c

Каждый верный ответ оценивается в 2 балла.

B1. Что будет выведено на экран в результате выполнения следующего участка кода, если на вход были поданы числа 15,16,17? Приведите пример входных данных, при которых на экран будет выведено число 3.

```
a,b,c=(int(input()) for i in range(3))
print(max(a,b,c)%4)
```

B2. Ниже записаны две рекурсивные функции (процедуры): F и G. Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(11)?

```
Python def F(n):
if n > 0: G(n - 1)
def G(n):
print("*")
if n > 1: F(n - 3)
```

B3. На обработку поступает положительное целое число, не превышающее 109. Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0. Программист написал программу неправильно.

```
N = int(input())
sum = 0
while N > 0:
digit = N % 10
if digit < 7:
sum = sum + 1
N = N // 10
print(digit)
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 358.

2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт верный ответ.

3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки: 1) выпишите строку, в которой сделана ошибка; 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Оценочный лист к типовому заданию B

B1. 1; 0,0,3

Программа выдаёт остаток от целочисленного деления максимального из числе a,b и c на 4. В примере это 17 и результат целочисленного деления - 1. Подходящим ответом на вторую часть задания будет любой набор чисел, в котором максимальное число при делении на 4 в остатке дает 3.

B2. 3

B3. Программа работает неправильно из-за неверной выводимой на экран переменной и неверного увеличения суммы. Соответственно, программа будет работать верно, если в числе старшая цифра (крайняя левая) равна сумме цифр, меньших 7.

1. Программа выведет число 3.

2. Пример числа, при вводе которого программа выдаёт верный ответ: 862.

3. В программе есть две ошибки.

1) Неверное увеличение суммы. Строка с ошибкой: sum = sum + 1; Верное исправление: sum = sum + digit.

2) Неверный вывод ответа на экран. Строка с ошибкой: print(digit) Верное исправление: print(sum).

C1. Составьте эффективную по памяти и по времени программу, проверяющую есть ли в числе одинаковые цифры?

C2. В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передаётся положительное целое число – текущее показание прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь. Необходимо вычислить «бета-значение» серии показаний прибора – минимальное чётное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удаётся, ответ считается равным –1. Напишите программу для решения поставленной задачи.

C3. Приведите примеры языков поддерживающий функциональное, логическое и процедурное программирование. Опишите отличия любых двух парадигм.

Оценочный лист к типовому заданию C

C1. Пример верно решенной задачи:

```
If len(x)=len(set(x)):
print (‘нет одинаковых цифр’)
else:
print (‘есть одинаковые цифры’)
```

Допускаются вариации, не влияющие на конечный результат вычисления, и не увеличивающие код существенно.

C2. Пример верно решенной задачи:

```
n=int(input())
l=[float(input())for i in range(n)]
sum_max=0
imax=0;
for i in range(5,n):
    if l[i-5]>sum_max:
        imax=l[i-5];
    if  l[i]+imax>sum_max:
```

```
    sum_max=l[i]+imax;
print(sum_max)
```

Допускаются вариации, не влияющие на конечный результат вычисления, и не увеличивающие код существенно.

С3. Один из вариантов ответа:

Функциональное программирование: Logo, Haskell, Erlang, F#, Python

Логическое программирование: Prolog

Процедурное программирование: Паскаль, Python, Basic, Си, Go и др.

Процедурное программирование подразумевает последовательность изменений состояния программы, а переменные служат для хранения этого состояния. Функциональное программирование, наоборот, предусматривает последовательность действий над данными.

Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (индикаторы)	Наименование оценочного средства	Максимальное количество баллов	Всего баллов	Уровень освоения компетенции (в баллах)		
				Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
УК-1.1	Задание А 1-10	10	10	5-6	7-8	9-10
УК-1.2	Задание А 11-20	10	10	5-6	7-8	9-10
УК-1.3	Задание В 1	10	10	5-6	7-8	9-10
УК-1.4	Задание В 2-3	20	20	10-14	15-16	17-20
УК-1.5	Задание С	10	10	5-6	7-8	9-10

Полученное число баллов (30-60) выставляется в графу «Промежуточная аттестация» балльно-рейтинговой карты дисциплины.