

Контрольно-оценочные средства для проведения текущего контроля

по УП.3 Учебной практики (4 курс, 7 семестр 2026-2027 уч. г.)

Текущий контроль №1 (45 минут)

Форма контроля: Практическая работа (Информационно-аналитический)

Описательная часть: Практическая работа

Задание №1 (10 минут)

1. Скопировать электронный бланк «Титульный лист» в папку группы.
2. Создать личный шифр для оформления пояснительной записки по курсовому проектированию на тему «Автоматизированная информационная система предприятия» согласно списку классного журнала.
3. Заполнить документ «Титульный лист» по своим данным.

Задание 2

Используя теоретический материал, знания, полученные на учебных занятиях создать и оформить лист «Содержание», если пояснительная записка будет включать в себя следующие разделы и подразделы:

Введение

1 Теоретическая часть

1.1 Описание деятельности предприятия

1.2 Описание структуры предприятия

1.2 Обзор современных программных средств создания баз данных предприятия

2 Проектная часть

2.1 Описание средств автоматизации предприятия

2.1.1 Состав аппаратных средств предприятия

2.1.2 Состав программных средств предприятия

2.2 Обоснование выбора среды разработки автоматизированной информационной системы

2.3 Модель проектированной информационной системы

2.3.1 Описание входных данных

2.3.2 Описание выходных данных

2.4 Алгоритм функционирования автоматизированной информационной системы

2.5 Интерфейс автоматизированной информационной системы

2.6 Инструкция пользователя

2.7 Техника безопасности при работе на персональном компьютере

Заключение

Приложение А. Программный код

Список использованных источников

Задание 3

Ответить на контрольные вопросы:

- 1) Поясните требования оформления текстового документа проекта.
- 2) Чем (какими документами) регламентируются требования форматирования текстовых документов по проектированию?
- 3) Поясните правила оформления таблиц в текстовом документе.
- 4) Поясните правила формирования шифра студента для документов проекта.
- 5) Поясните требования оформления содержания пояснительной записки.
- 6) Перечислите (приведите пример) способы оформления списков.

Оценка	Показатели оценки
5	Выполнены все задания.
4	Выполнено 2 задания из 3.
3	Выполнено 1 задание из 3.

Задание №2 (10 минут)

- Решите задачи, используя только элементарные конструкции (последовательность, ветвления, циклы). Программа должна быть рабочей!
- Оптимизировать программу (можно использовать процедуры или функции).

Внимание! Оптимизированная программа должна содержать проверки всех переменных, которые вводятся с клавиатуры.

- Для созданных программ оценить метрические характеристики по Холстеду;
- Сравнить полученные результаты. Оформить результаты в таблицу. Сделать соответствующие выводы.

Задача 1. Даны натуральные числа $n, a_1, \dots, a_n$. Определить количество членов a_k последовательности $a_1, \dots, a_n$:

- являющихся нечетными числами;
- кратных 3 и не кратных 5;
- имеющих четные порядковые номера и являющихся нечетными числами.

Задача 2. Даны натуральные числа $n, q_1, \dots, q_n$. Найти те члены q_i последовательности $q_1, \dots, q_n$, которые

- являются удвоенными нечетными числами;
- при делении на 7 дают остаток 1, 2 или 5;
- делятся на 5 и не делятся на 7, найти количество и сумму тех членов последовательности.

Задача 3. Даны целые числа $A_1, \dots, A_{80}$. Получить сумму тех чисел данной последовательности, которые

- кратны 5;
- нечетны и отрицательны;
- удовлетворяют условию $A[i]$.

Оценка	Показатели оценки
5	Выполнены все задания.
4	Выполнено 2 задания из 3.
3	Выполнено 1 задание из 3.

Выполнение №3 (5 минут) Выберите любую программу (выполненную ранее), для которой имеются исходные тексты, и определите ее сложность согласно метрикам размера программ, сложности потока управления программ, сложности потока данных.

Оценка	Показатели оценки
5	Задание выполнено верно, определена сложность согласно метрикам размера программ, сложности потока управления программ, сложности потока данных.
4	Задание выполнено верно, определена сложность согласно метрикам размера программ, сложности потока управления программ.
3	Задание выполнено верно, определена сложность согласно метрикам размера программ.

Выполнение №4 (10 минут) Проведите анализ средств разработки программных продуктов, для разработки информационных систем, с целью выявления наилучшего решения согласно критериям:

- выполнить анализ достоинств и недостатков программных продуктов;
- обосновать выбор одного (возможно, двух и более) из них;
- оформить в виде сравнительной таблицы.

Оценка	Показатели оценки
5	Задание выполнено верно, соблюдены все критерии.
4	Задание выполнено верно, соблюдено 2 критерия из 3.
3	Задание выполнено верно, соблюдено 1 критерий из 3.

Выполнение №5 (10 минут) Проведите (code review) небольшого фрагмента исходного кода.

Задача: 1. Внимательно изучите предоставленный код.

2. Проведите ревью кода, обращая внимание на следующие аспекты: - Читаемость и стиль кода (соблюдение стандартов кодирования, названия переменных, комментарии). - Корректность алгоритмов и логики кода. - Обработка ошибок и исключительных ситуаций. - Возможные уязвимости и потенциальные ошибки. - Производительность и оптимизация кода.

3. Сформулируйте замечания и рекомендации по улучшению кода.

4. Подготовьте краткий отчет с результатами ревью, включающий: - Список обнаруженных

проблем или недочетов в коде и документации. - Рекомендации по исправлению или улучшению кода и документации. - Общую оценку качества кода и документации.

Программный код:

Модуль для работы с банковскими счетами.

Этот модуль предоставляет функциональность для создания и управления банковскими счетами.

class BankAccount: Класс для представления банковского счета.

Атрибуты:account_number (str): Номер банковского счета. balance (float): Текущий баланс счета.

```
def __init__(self, account_number):
```

Инициализирует объект BankAccount.

Args:

account_number (str): Номер банковского счета.

```
self.account_number = account_number
```

```
self.balance = 0.0
```

```
def deposit(self, amount):
```

Пополняет баланс счета.

Args:

amount (float): Сумма для пополнения счета.

Raises:

ValueError: Если сумма пополнения отрицательная.

```
if amount < 0:
```

```
raise ValueError("Сумма пополнения должна быть положительной.")
```

```
self.balance += amount
```

```
def withdraw(self, amount):
```

Снимает указанную сумму со счета.

Args:

amount (float): Сумма для снятия со счета.

Raises:

ValueError: Если сумма снятия отрицательная.

ValueError: Если на счете недостаточно средств.

```
if amount < 0:
```

```
raise ValueError("Сумма снятия должна быть положительной.")
```

```
if self.balance < amount:
```

```
raise ValueError("Недостаточно средств на счете.")
```

```
self.balance -= amount
```

```
def get_balance(self):
```

Возвращает текущий баланс счета.

Returns:

float: Текущий баланс счета.

return self.balance

Оценка	Показатели оценки
5	Студен успешно справился с заданием. Провел ревью кода, учел все аспекты. Сформулировал замечания и рекомендации по улучшению кода. Подготовил краткий отчет с результатами ревью, включающий: - Список обнаруженных проблем или недочетов в коде. - Рекомендации по исправлению или улучшению кода. - Общую оценку качества кода.
4	Студен справился с заданием. Провел ревью кода, учел все аспекты. Сформулировал замечания и рекомендации по улучшению кода. Подготовил краткий отчет с результатами ревью, включающий: - Список обнаруженных проблем или недочетов в коде.
3	Студент провел ревью кода, учел все аспекты. Сформулировал замечания и рекомендации по улучшению кода.

Текущий контроль №2 (45 минут)

Форма контроля: Практическая работа (Сравнение с аналогом)

Описательная часть: Практическая работа

Задание №1 (10 минут) программу на Python. Даны длины сторон треугольника, определить вид треугольника и его площадь. Выполнить контроль вводимых чисел.

1. Разносторонний треугольник.

2. Равнобедренный треугольник.

3. Равносторонний треугольник.

Ограничения:

- три числа не могут быть определены как стороны треугольника;
- если хотя бы одно из них меньше или равно 0;
- сумма двух из них меньше третьего.

Задание 2. Подготовить набор тестовых вариантов для обнаружения ошибок в программе.

Результат оформить в следующем виде:

А	И	С	Ожидаемый результат	Объект проверки
Значение	Значение	Значение	Что должно получиться	Значения вводимых данных, либо ожидаемый результат

Сделать вывод о проделанной работе.

Оценка	Показатели оценки
5	Выполнены все задания, сделан вывод о проделанной работе.
4	Выполнены все задания, отсутствует вывод.
3	Выполнены два задания.

Задача №2 (10 минут) - анализе метрик программного проекта, чтобы получить представление о его характеристиках, таких как размер, сложность и качество кода.

Задача: 1. Изучить исходный код программного проекта.

2. Запустите инструмент анализа метрик на предоставленном исходном коде проекта.

3. Проанализируйте следующие метрики: - Размер проекта (количество строк кода, количество файлов, количество классов/функций). - Цикломатическая сложность (complexity). - Покрытие кода тестами (если доступно). - Количество дублированного кода. - Количество потенциальных ошибок, уязвимостей и проблем с качеством кода (согласно правилам анализатора).

4. Оцените полученные метрики и сделайте выводы о характеристиках проекта: - Является ли размер проекта большим или маленьким? - Какова сложность кода в среднем? - Достаточно ли хорошее покрытие кода тестами? - Много ли дублированного кода в проекте? - Какие потенциальные проблемы с качеством кода были выявлены?

5. Подготовьте краткий отчет с результатами анализа метрик, включающий: - Список анализируемых метрик и их значения. - Выводы о характеристиках проекта на основе полученных метрик. - Рекомендации по улучшению проекта (если необходимо).

Программный код:

Модуль для работы с калькулятором.

Этот модуль предоставляет функциональность для выполнения основных арифметических операций с двумя числами.

```
def add(a, b): // Сложение двух чисел.
```

Args:

a (float): Первое число.

b (float): Второе число.

Returns:
float: Результат сложения.
return a + b

def subtract(a, b):
Вычитание двух чисел.

Args:
a (float): Уменьшаемое.
b (float): Вычитаемое.

Returns:
float: Результат вычитания.
return a - b

def multiply(a, b):
Умножение двух чисел.

Args:
a (float): Первый множитель.
b (float): Второй множитель.

Returns:
float: Результат умножения.
return a * b

def divide(a, b):
Деление двух чисел.

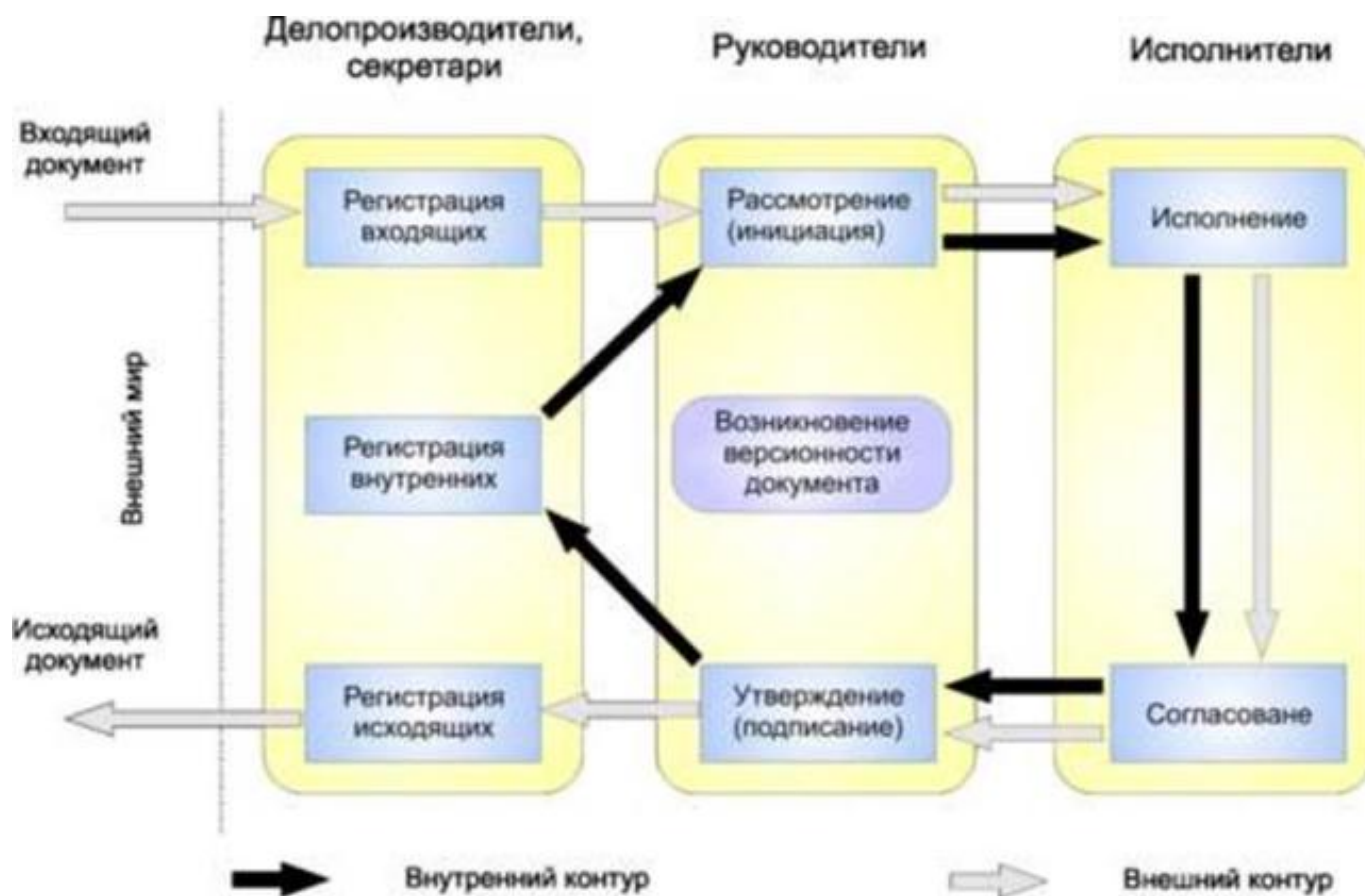
Args:
a (float): Делимое.
b (float): Делитель.

Raises:
ZeroDivisionError: Если делитель равен нулю.

Returns:
float: Результат деления.
if b == 0:
raise ZeroDivisionError("Деление на ноль невозможно.")
return a / b

Оценка	Показатели оценки
5	Студент успешно выполнил пять поставленных перед ним задач.
4	Студент успешно выполнил четыре поставленных перед ним задач.
3	Студент успешно выполнил три поставленных перед ним задач.

Задание №3 (10 минут) Функциональные и нефункциональные требования к программному обеспечению для документооборота. Вы должны представить свое видение таких требований, если схема документооборота выглядит следующим образом:



Оценка	Показатели оценки
5	Представлено не менее 4 функциональных и нефункциональных требований. Сделан вывод о проделанной работе.
4	Представлено не менее 3 функциональных и нефункциональных требований. Сделан вывод о проделанной работе.
3	Представлено менее 3 функциональных и нефункциональных требований.

Задание №4 (10 минут) Какие методологии разработки ПО, как: Scrum, Agile, Kanban. Оформите в виде таблицы. Сделайте вывод по проделанной работе.

Оценка	Показатели оценки
5	Методологии рассмотрены по более чем 4 критериям, сделан вывод о проделанной работе.
4	Методологии рассмотрены менее чем по 5 критериям, сделан вывод о проделанной работе.
3	Методологии рассмотрены менее чем по 5 критериям.

Задание №5 (5 минут)
 Программа: У кассы Аэрофлота выстроилась очередь из N человек. Время обслуживания кассиром i-го клиента равно T_i ($i = 1, \dots, N$). 1. Определите время пребывания в очереди каждого клиента 2. Укажите номер клиента, для обслуживания которого кассиру потребовалось больше всего времени.

Задание 2. Провести функциональное тестирование разработанной программы.

Задание 3. Провести тестирование методом "черного ящика".

Задание 4. Провести тестирование "белого ящика".

Оценка	Показатели оценки
5	Выполнены все задания.
4	Выполнено 3 задания из 4.
3	Выполнено 2 задания из 4.

Текущий контроль №3 (45 минут)

Форма контроля: Практическая работа (Сравнение с аналогом)

Описательная часть: Практическая работа

Задание №1 (10 минут)

1. Оценить программу по следующим метриками:

- количество строк исходного текста (под строкой понимается любой оператор программы);
- метрики Холстеда – длины, объема, потенциального объема, уровня реализации, уровня программы, интеллектуального содержания, оценка необходимых интеллектуальных усилий по написанию программы;
- цикломатическое число Мак-Кейба;
- метрика Джилба;
- метрика обращения к глобальным переменным;
- метрика Спена;
- метрика Чепина;
- метрика уровня комментированности программы.

2. Ответить на контрольные вопросы.

3. Подготовить отчет с результатами оценок программы.

Ответить на контрольные вопросы:

1. Что такое критерий качества? Перечислите его основные характеристики.
2. Что такое метрика качества программы?
3. Какие два основных направления исследования метрик ПО существуют?
4. На какие три группы делятся метрики виду информации, получаемой при оценке качества ПО?
5. Какие метрические шкалы существуют?
6. Какие основные группы метрик выделяют при оценке сложности программ?

Оценка	Показатели оценки
5	Выполнены все задания.
4	Выполнено 2 задания из 3.
3	Выполнено 1 задание из 3.

Задание №2 (10 минут)
Задание 1. Определите соответствие следующих нормативных документов:

ГОСТ Р 54869-2011 Проектный менеджмент. Требования к управлению проектом.

ГОСТ Р 54870-2011 Проектный менеджмент. Требования к управлению портфелем проектов.

ГОСТ Р 54871-2011 Проектный менеджмент. Требования к управлению программой.

Задание 2. Для формирования перечня документов и записей, требуемых стандартом ГОСТ Р 54869-2011, определите необходимые документы применительно к разрабатываемому проекту и найдите записи, обеспечивающие объективные доказательства выполнения каждого требования.

Задание 3. Результаты работы необходимо занести в соответствующие разделы таблицы 1.

Таблица 1

Раздел, пункт, подпункт стандарта	Наименование требования стандарта	Наименование документов или записей по качеству, подтверждающих выполнение требования стандарта
Процесс инициации проекта		
Процессы планирования проекта		
Процесс организации		

исполнения проекта		
Процесс контроля исполнения проекта		
Процесс завершения проекта		
Требования к управлению документами проекта		

Оценка	Показатели оценки
5	Выполнены все задания.
4	Выполнено 2 задания из 3.
3	Выполнено 1 задания из 3.

Задание №3 (10 баллов) Какие методологии разработки ПО, как: Scrum, Agile, Kanban. Оформите в виде таблицы. Сделайте вывод по проделанной работе.

Оценка	Показатели оценки
5	Методологии рассмотрены по более чем 4 критериям, сделан вывод о проделанной работе.
4	Методологии рассмотрены менее чем по 5 критериям, сделан вывод о проделанной работе.
3	Методологии рассмотрены менее чем по 5 критериям.

Задание №4 (10 баллов) Условный код, методом Constant folding (Свертывание/Свертка констант) и методом Constant propagation (Распространение констант):

```
#include <stdlib.h>
```

```
int main(int argc, char **argv)
```

```
{struct point
```

```
{
```

```
int x;
```

```
int y;
```

```
} p;
```

```
int a = 32*32;
```

```

int b = 32*32*4;

long int c;

//

c = (a + b) * (4*4*sizeof(p) - 2 + 32);

return 0;}

```

Сделайте вывод о проделанной работе.

Оценка	Показатели оценки
5	Задание выполнено верно. Сделан вывод о проделанной работе.
4	Оптимизирован код только по одному методу. Сделан вывод о проделанной работе.
3	Оптимизирован код только по одному методу.

Выполнение №5 (5 минут) диаграммы прецедентов и диаграмму классов UML по теме «ИС Гостиница».

Оценка	Показатели оценки
5	Разработано более 2 диаграмм прецедентов. Разработана диаграмма классов. Диаграммы описаны.
4	Разработана диаграмма прецедентов. Разработана диаграмма классов. Диаграммы описаны.
3	Разработана диаграмма прецедентов. Разработана диаграмма классов.