

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Ухтинский государственный технический университет»
(УГТУ)
филиал Ухтинского государственного технического университета
в г. Усинске
(УФ УГТУ)
(среднего профессионального образования)

УТВЕРЖДАЮ
И. о. директора филиала
_____ О. В. Филиппова

« ____ » _____ 20__ г.

КОМПЛЕКС ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
МДК 02.02. Инструментальные средства разработки
программного обеспечения
образовательной программы
среднего профессионального образования
По специальности 09.02.11 Разработка и управление программным
обеспечением

г. Усинск
2026

1. ПАСПОРТ КОМПЛЕКТА ОЦЕНОЧНЫХ СРЕДСТВ

1.1 Область применения

Комплект оценочных средств (далее – КОС) предназначен для контроля и оценки результатов прохождения учебной дисциплины МДК 02.02. «Инструментальные средства разработки программного обеспечения», образовательной программы среднего профессионального образования (далее – ОП СПО) по специальности 09.02.11 «Разработка и управление программным обеспечением».

1.2 Результаты освоения компетенции

В результате проведения промежуточной аттестации по дисциплине осуществляется комплексная оценка овладения следующими профессиональными и общими компетенциями:

Код	Результат освоения компетенции
ПК 2.1	Проектировать модули программного обеспечения.
ПК 2.2	Разрабатывать модули программного обеспечения.
ПК 2.3	Выполнять интеграцию модулей и компонентов программного обеспечения.
ПК 2.4	Выполнять тестирование и отладку программного обеспечения
ПК 2.5	Осуществлять документирование программных модулей программного обеспечения.
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языках

2. ФОРМЫ КОНТРОЛЯ И ОЦЕНКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В соответствии с учебным планом и рабочей программой предусматривается текущий и промежуточный контроль результатов освоения.

Предметом оценки служат умения и знания, предусмотренные требованиями ФГОС по дисциплине МДК 02.02. «Инструментальные средства разработки программного обеспечения», направленные на формирование общих и профессиональных компетенций, а также личностных результатов в рамках программы воспитания.

2.1 Критерии и нормы оценки знаний обучающихся.

Оценка «5» ставится в случае:

- знания, понимания, глубины усвоения обучающимися всего объёма программного материала;
- умения выделять главные положения в изученном материале, на основании фактов и примеров обобщать, делать выводы, устанавливать межпредметные и внутри предметные связи, творчески применять полученные знания в незнакомой ситуации;
- отсутствия ошибок и недочётов при воспроизведении изученного материала, при устных ответах устранения отдельных неточностей с помощью дополнительных вопросов учителя, соблюдения культуры письменной и устной речи, правил оформления письменных работ.

Оценка «4» ставится в случае:

- знания всего изученного программного материала;
- умения выделять главные положения в изученном материале, на основании фактов и примеров обобщать, делать выводы, устанавливать внутри предметные связи, применять полученные знания на практике;
- допущения незначительных (негрубых) ошибок, недочётов при воспроизведении изученного материала; соблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Оценка «3» ставится в случае:

- знания и усвоения материала на уровне минимальных требований программы, затруднения при самостоятельном воспроизведении, возникновения необходимости незначительной помощи преподавателя;
- умения работать на уровне воспроизведения, затруднения при ответах на видоизменённые вопросы;
- наличия грубой ошибки, нескольких грубых ошибок при воспроизведении изученного материала; незначительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Оценка «2» ставится в случае:

- знания и усвоения материала на уровне ниже минимальных требований программы; наличия отдельных представлений об изученном материале;
- отсутствия умения работать на уровне воспроизведения изученного материала, затруднения при ответе на стандартные вопросы;
- наличия нескольких грубых ошибок, большого числа негрубых при воспроизведении изученного материала, значительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

2.2 Критерии оценки за тестирование

Оценка 5 «отлично» (85 % -100 %)

Оценка 4 «хорошо» (75 % - 84 %)

Оценка 3 «удовлетворительно» (55 % -74 %)

Оценка 2 «неудовлетворительно» (менее 54 %)

2.3 Формы и методы оценивания учебной дисциплины «Инструментальные средства разработки программного обеспечения»

Раздел, тема	Формы и методы контроля					
	Текущий контроль		Промежуточная аттестация		Итоговая аттестация	
	Формы контроля	Проверяемые Знания, умения, ОК, ЛР	Формы контроля	Проверяемые знания, умения ОК, ЛР	Формы контроля	Проверяемые ЗУН, ОК, ЛР
Тема1. Современные инструменты планирования и проектирования ПО	ПР№1, УО.	ПК2.2, ПК2.3, ПК2.5.	ДФК, Т	ПК2.2, ПК2.3, ПК2.5.	Эк	ПК2.2, ПК2.3, ПК2.5.
Тема2. Современные технологии и инструменты интеграции.	ПР№2, ПР№3, УО, СР	ПК2.2, ПК2.3, ПК2.5.	ДФК, Т	ПК2.2, ПК2.3, ПК2.5.	Эк	ПК2.2, ПК2.3, ПК2.5.
Тема 3 Инструментарий тестирования и анализа качества программных средств	ПР№4, УО.	ПК2.2, ПК2.3, ПК2.5.			Эк	ПК2.2, ПК2.3, ПК2.5.

Кодификатор оценочных средств

Функциональный признак оценочного средства (тип контрольного задания)	Код оценочного средства
Устный опрос	УО
Практическая работа	ПР №
Самостоятельная работа	СР
Тестирование	Т
Другая форма контроля	ДФК
Экзамен	Эк

3. Контрольно-оценочные материалы для промежуточной и итоговой аттестации по учебной дисциплине «Инструментальные средства разработки программного обеспечения»

3.1 Типовые задания для проведения текущего контроля

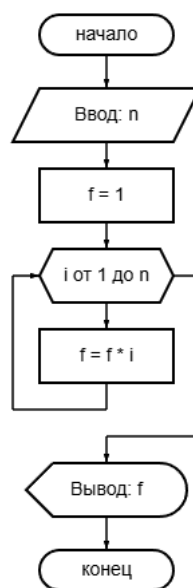
3.1.1 Перечень практических работ.

Практическая работа №1:

Задача 1. Спроектировать блок – схему ПО используя возможности MS Visio или его аналог.

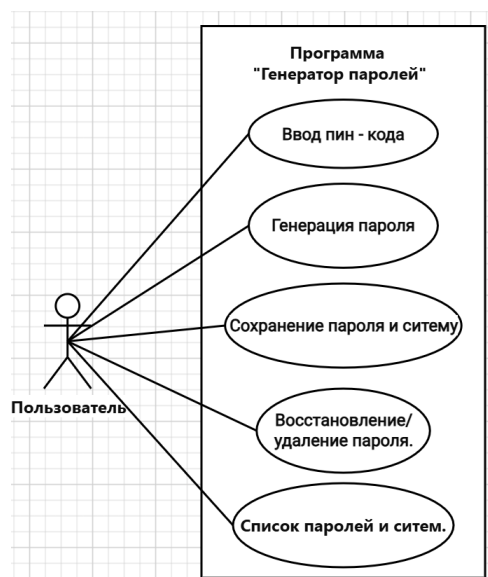
Пример № 1. Сформируйте блок – схему для вычисления факториала натурального числа n .

Решение:



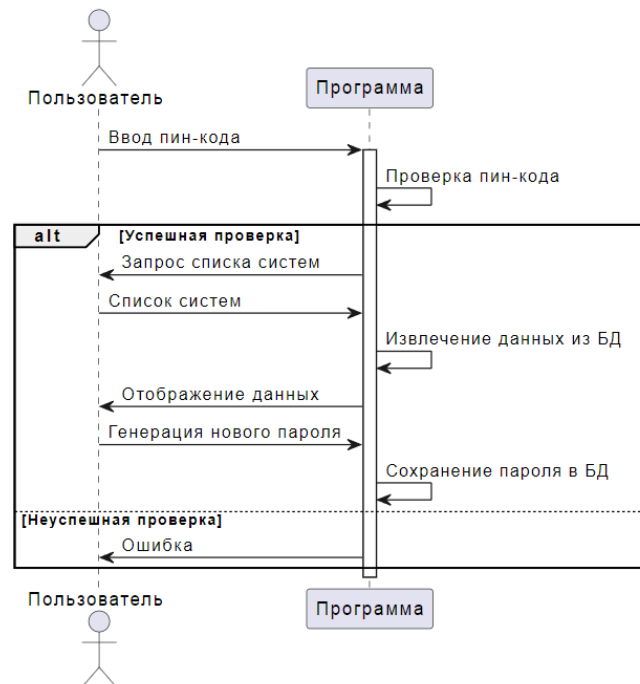
Задача 2. Спроектировать «Use case» диаграмму (диаграмму вариантов использования) для программы, которая генерирует и запоминает пароли к той или иной системе. Пользователь может видеть список систем и паролей. Вход в программу осуществляется по пин – коду.

Решение:



Задача 3. Спроектировать Sequence диаграмму (диаграмму последовательности), для программы, которая генерирует и запоминает пароли к той или иной системе. Пользователь может видеть список систем и паролей. Вход в программу осуществляется по пин – коду.

Решение:



Практическая работа №2:

Задача № 1

1. Определите, что выполняет программа.
2. Выявите ошибки программы и исправьте их.
3. Укажите, какой вид ошибки допущено в задаче.

```

#include <iostream>
using namespace std;
int main()
{
    double a, b, c , d;
    double s, sr;
    int n = 4;

    cout >> "a = ";
    cin << a;
    cout >> "b = ";
    cin << b;
    cout >> "c = ";
    cin << c;
    cout >> "d = ";
    cin << d;

    s = a + b + c + d;
    cout << "s = " << s << endl;

    Sr = s / 4;
    cout << "sr = " << Sr << endl;
}
  
```

Решение:

1. Программа рассчитывает сумму введенных чисел и их среднее арифметическое значение.

2. Исправленное ПО:

В программе были выявлены следующие ошибки

Исходный код	Сообщение
abc E0349	отсутствует оператор ">>", соответствующий этим операндам
abc E0349	отсутствует оператор ">>", соответствующий этим операндам
abc E0349	отсутствует оператор ">>", соответствующий этим операндам
abc E0349	отсутствует оператор ">>", соответствующий этим операндам
abc E0349	отсутствует оператор "<<", соответствующий этим операндам
abc E0349	отсутствует оператор "<<", соответствующий этим операндам
abc E0349	отсутствует оператор "<<", соответствующий этим операндам
abc E0349	отсутствует оператор "<<", соответствующий этим операндам
abc E0020	идентификатор "Sr" не определен

1. Не верное направление стрелок операторов cout и cin.

2. Имя для переменной sr используется в различных регистрах. А язык программирования C++ регистрочувствительный.

```
#include <iostream>
using namespace std;
int main()
{
    double a, b, c, d;
    double s, sr;
    int n = 4;

    cout << "a = ";
    cin >> a;
    cout << "b = ";
    cin >> b;
    cout << "c = ";
    cin >> c;
    cout << "d = ";
    cin >> d;

    s = a + b + c + d;
    cout << "s = " << s << endl;

    sr = s / 4;
    cout << "sr = " << sr << endl;
}
```

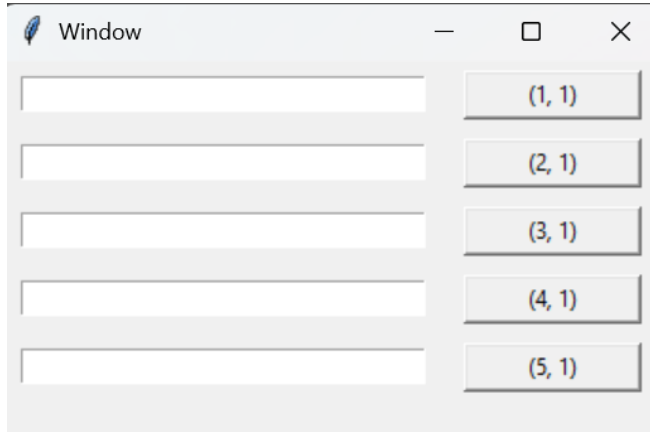
```
a = 2
b = 3
c = 5
d = 1
s = 11
sr = 2.75
```

3. Все ошибки синтаксические.

Практическая работа №3.

Задача № 1

Создайте окно, в котором перечисляются 5 подряд однострочный ввод в нулевой колонке и кнопки в первой колонке, с помощью метода **grid()**.



Решение:

```
import tkinter as tk

root = tk.Tk()

root.title("Window")
root.geometry("350x200")

for i in range(5):
    # Поле ввода (колонка 0)
    entry = tk.Entry(root, width=35)
    entry.grid(row=i, column=0, padx=10, pady=5)

    # Кнопка (колонка 1)
    btn = tk.Button(root, text=f" ({i+1}, 1) ", width=12)
    btn.grid(row=i, column=1, padx=10, pady=5)

root.mainloop()
```

Практическая работа №4.

Задача № 1. Необходимо проанализировать программу и устранить ошибки. Выполнить тестирование исправленной программы.

```
#include <iostream>
#include <locale>
#include <vector>
#include <math.h>
#include <memory>

using namespace std;

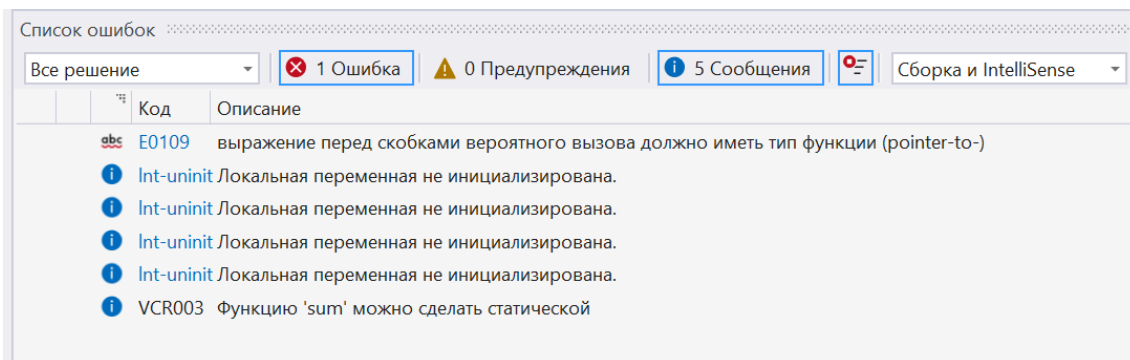
int sum(int a, int b)
{
    return a + b;
}

int main()
{
    setlocale(LC_CTYPE, "rus");
    int n, s, c, b;
    double ms[4];
    int* a; // указатель на первый элемент
    cout << "Введите количество элементов массива n: ";
    cin >> n;
    while (n <= 0)
    {
        cout << "Введите положительное число n: ";
        cin >> n;
    }
    //Выделение памяти для массива размера n
    a = new int[n];
    while (a == NULL)
    {
        cout << "Не возможно выделить такое количество памяти" << endl;
        cout << "Попробуйте ввести меньшее количество: ";
        cin >> n;
    }
    cout << "Введите элементы массива:" << endl;
    for (int i = 0; i < n; i++)
        cin >> a[i];
    int sum = 0;
    for (int i = 0; i < n; i++)
    {
        sum += a[i];
    }
    for (int i = 0; i < n; i++)
    {
        cout << ms[i];
    }

    cout << "function sum = " << sum(233, 123) << endl;

    cout << "array sum = " << sum << endl;
    // освобождение памяти, занимаемой массивом
    delete[] a;
    return 0;
}
```

Решение:

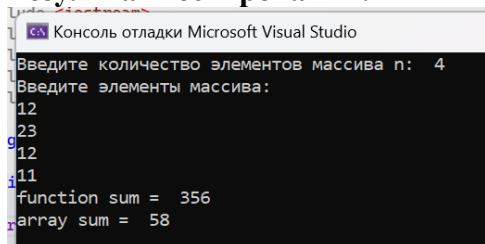


1. Ошибка в программе вызвана в связи с тем, что имя функции совпадает с именем переменной. Необходимо поменять имя функции.

2. Причина сообщений:

- переменные s, c, b были объявлены, но не используются.
- был объявлен массив ms[4] без инициализации, который в дальнейшем выводится на экран. Массив не используется в программе, его следует удалить.
- функция sum следует сделать статическим.

Результат тестирования:



3.1.2. Примерные вопросы для устного опроса обучающихся по учебной дисциплине

Тема 1. Современные инструменты планирования и проектирования ПО

- 1.1. Для каких целей используют диаграмму Ганте?
- 1.2. Что такое блок – схема. Какие программные средства используют для создания блок – схемы.?
- 1.3. Какие основные блоки используют для формирования блок - схем?
- 1.4. Что такое UML?
- 1.5. Объясните назначения диаграммы сценариев использования?
- 1.6. Объясните назначения диаграммы последовательностей?

Тема 2. Современные технологии и инструменты интеграции.

- 2.1. Что значит интегрированная среда разработки (IDE, Integrated Development Environment).
- 2.2. Приведите примеры интегрированной среды разработки.
- 2.3. Что такое система контроля версий?
- 2.4. Назовите основные команды Git?
- 2.5. Что такое СУБД?
- 2.6. Каким способом можно создать таблицу в MS Access?
- 2.7. Для чего предназначено ключевое поле в MS Access?
- 2.8. С какой целью, используют фильтрацию данных в MS Access?
- 2.9. Что такое форма в MS Access

- 2.10. Основные задачи запросов в СУБД MS Access?
- 2.11. Основные типы данных SQL?
- 2.12. Для создания таблицы в SQL, что необходимо указать в запросе?

Тема 3. Инструментарий тестирования и анализа качества программных средств

- 3.1. Назовите типы сообщений компилятора в MS Visual Studio?
- 3.2. Что такое трассировка программы?
- 3.3. Как использовать точки останова в MS Visual Studio?

3.1.3 Перечень самостоятельных работ.

Задача № 1.

Перечислите основные команды для создания и просмотра истории коммитов в Git.

Решение:

Команда	Описание
git commit -m "<комментарий к коммиту>"	Создание коммита с сообщением
git log	Просмотр истории коммитов

3.2. Типовые задания для проведения промежуточной аттестации

Другая форма контроля

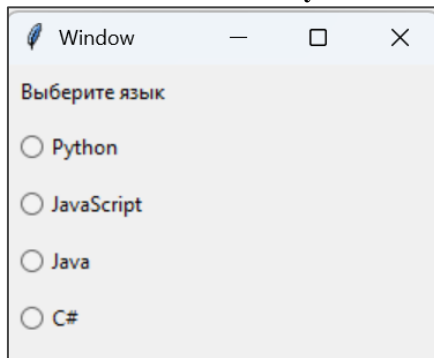
- 1) Средства разработки программного обеспечения – это**
 - a) совокупность приемов, методов, методик, а также набор инструментальных программ, используемых разработчиком для создания программного кода, отвечающего заданным требованиям.**
 - b) сложный процесс, основной целью которого является создание, сопровождение программного кода, обеспечивающего необходимый уровень надежности и качества.
 - c) совокупность программ, баз данных, файлов, а также описывающих их документов, составляющих систему для решения группы связанных задач на одном или нескольких взаимодействующих компьютерах.
 - d) компьютерная программа или логически связанная совокупность программ, предназначенная для автоматизации в определённой области профессиональной деятельности.
- 2) Разработка программ – это**
 - a) совокупность приемов, методов, методик, а также набор инструментальных программ, используемых разработчиком для создания программного кода, отвечающего заданным требованиям.
 - b) сложный процесс, основной целью которого является создание, сопровождение программного кода, обеспечивающего необходимый уровень надежности и качества.**
 - c) совокупность программ, баз данных, файлов, а также описывающих их документов, составляющих систему для решения группы связанных задач на одном или нескольких взаимодействующих компьютерах.
 - d) компьютерная программа или логически связанная совокупность программ, предназначенная для автоматизации в определённой области профессиональной деятельности.
- 3) На каком этапе разработки ПО выполняется анализ требования, на основе которой создается содержательная или формализованная модель процесса функционирования программы:**
 - a) **Проектирование ПО**
 - b) Реализация программного кода приложения
 - c) Тестирование ПО
- 4) На этом этапе производят проверки соответствия функциональности разработанной программы первоначальным требованиям, а также выявление ошибок, которые в явном или неявном виде проявляются во время работы программы.**
 - a) Проектирование ПО
 - b) Реализация программного кода приложения
 - c) **Тестирование ПО**
- 5) Специально ПО, которое преобразует сразу весь исходный текст в двоичный код:**
 - a) **Компилятор**
 - b) Интерпретатор
 - c) Компоновщик
- 6) Специально ПО, которое преобразует текст на языке высокого уровня построчно. Как только первая строка преобразована - она тут же запускается на выполнение:**
 - a) Компилятор
 - b) **Интерпретатор**
 - c) Компоновщик
- 7) Инструмент управления проектами, который позволяет регулировать работу по нескольким направлениям: планировать и составлять графики проектов, отслеживать зависимости между задачами, делегировать, распределять ресурсы, постоянно держать под контролем процесс выполнения каждой задачи:**

- a) Диаграмма последовательностей
 - b) UML диаграмма
 - c) **Диаграмма Ганта**
- 8) **Диаграмма, которая описывает, какой функционал разрабатываемой системы доступен каждой группе пользователей:**
- a) диаграммы последовательностей
 - b) диаграмма объектов
 - c) **диаграммы сценариев использования**
- 9) **Диаграмма UML – это**
- a) **унифицированный язык моделирования, которую используют, чтобы создавать диаграммы и схемы для визуализации процессов и явлений.**
 - b) способ графического представления алгоритма, в котором шаги изображаются в виде блоков различной формы, соединенных между собой стрелками.
 - c) графическое представление данных линейными отрезками или геометрическими фигурами, позволяющее быстро оценить соотношение нескольких величин.
- 10) **Система контроля версий (СКВ) – это**
- a) Хранилище файлов, предназначенное для организации совместной работы программистов по созданию какой-либо программы.
 - b) совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.
 - c) **система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.**
- 11) **Виджет библиотеки Tkinter для однострочного ввода данных:**
- a) label
 - b) **entry**
 - c) text
- 12) **Виджет библиотеки Tkinter для вывода данных на диалоговое окно:**
- a) **label**
 - b) entry
 - c) text
- 13) **Виджет библиотеки Tkinter, реализующий кнопку:**
- a) radiobutton
 - b) checkbutton
 - c) **button**
- 14) **Самый простой способ размещения, с помощью которого виджеты "упаковываются" друг за другом:**
- a) place()
 - b) grid()
 - c) **pack()**
- 15) **Размещает виджеты в таблице (сетке):**
- a) place()
 - b) **grid()**
 - c) pack()
- 16) **Абсолютное позиционирование (пиксель):**
- a) **place()**
 - b) grid()
 - c) pack()
- 17) **Какой виджет, позволяет выполнить чертеж графика функции:**
- a) **canvas**
 - b) notebook
 - c) frame

18) С помощью какого виджета можно создать вкладки в диалоговом окне:

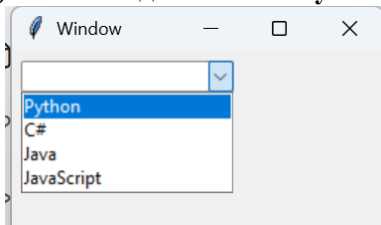
- a) canvas
- b) **notebook**
- c) frame

19) Какой виджет используется на картинке?



- a) **radiobutton**
- b) checkbutton
- c) button

20) Какой виджет используется на картинке?



- a) radiobutton
- b) checkbutton
- c) **combobox**

3.2 Организация и проведение итоговой аттестации

3.2.1 Условия проведения итоговой аттестации. Подготовка к проведению итоговой аттестации.

Итоговая аттестация проводится в период экзаменационной сессии, установленной календарным графиком учебного процесса рабочего учебного плана. С формами проведения промежуточной аттестации обучающиеся знакомятся в течение двух месяцев с начала обучения.

Для подготовки к итоговой аттестации студентом (не позднее чем за 20 дней до проведения экзамена в соответствии с календарным графиком учебного процесса) выдается перечень тем к практическим заданиям, составленный исходя из требований ФГОС и рабочей программы предмета к уровню умений и знаний. Количество вопросов и практических задач в перечне для подготовки к промежуточной аттестации превышает количество вопросов и практических задач, необходимых для составления контрольно-измерительных материалов.

На основе разработанного и объявленного обучающимся перечня вопросов и практических задач, рекомендуемых для подготовки к итоговой аттестации, составлены задания, содержание которых до обучающихся не доводится. Вопросы и практические задачи носят равноценный характер. Формулировки вопросов четкие, краткие, понятные, исключают двойное толкование.

3.2.2 Проведение итоговой аттестации

Студенты для сдачи итоговой аттестации распределяются по времени. На выполнение задания студенту отводится не более двух академических часов. Оценка, полученная на итоговой аттестации, заносится преподавателем в зачетную книжку студента (кроме неудовлетворительной) и экзаменационную ведомость (в том числе и неудовлетворительные). Экзаменационная оценка по предмету за данный семестр является определяющей, независимо от полученных в семестре оценок текущего контроля по предмету

ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Теоретические вопросы:

1. Планирование IT проекта. Диаграмма Ганта.
2. Блок - схема. Основные элементы блок – схемы.
3. Интегрированная среда разработки. Привести примеры ИСР.
4. Дайте определение отладке и тестированию. Перечислите виды программных ошибок.
5. Диаграмма UML. Диаграммы сценариев использования.
6. Диаграмма UML. Диаграммы последовательностей.
7. Диаграмма UML. Диаграмма потоков данных.
8. Компилятор и интерпретатор. Перечислите сходство и отличие.
9. Какие среды разработки существуют для языка программирования Python.
10. Какие средства разработки используют для создания пользовательского интерфейса (язык программирования Python)?
11. Среда разработки Visual Studio Code. Терминал, основные команды.
12. Виртуальная среда. Главные преимущества виртуальных сред.
13. Виртуальная среда. Основные команды для создания и запуска ВС через менеджера зависимостей.
14. Библиотека Tkinter. Преимущества и недостатки.
15. Библиотека Tkinter. Что такое ttk?
16. Позиционирование pack(). Перечислите и опишите основные параметры.
17. Позиционирование grid(). Перечислите и опишите основные параметры.
18. Позиционирование place(). Перечислите и опишите основные параметры.
19. Назначение виджета button. Перечислите и опишите основные параметры.
20. Назначение виджета label. Перечислите и опишите основные параметры.
21. Назначение виджета entry. Перечислите и опишите основные параметры.
22. Назначение виджета checkbox. Перечислите и опишите основные параметры.
23. Назначение виджета radiobutton. Перечислите и опишите основные параметры.
24. Назначение виджета frame. Перечислите и опишите основные параметры.
25. Назначение виджета notebook. Перечислите и опишите основные параметры.
26. Назначение виджета listbox. Перечислите и опишите основные параметры.
27. Назначение виджета scrollbar. Перечислите и опишите основные параметры.
28. Назначение виджета combobox. Перечислите и опишите основные параметры.
29. Назначение виджета canvas. Перечислите и опишите основные параметры.
30. Окна сообщений. Перечислите и опишите основные параметры.
31. Окна подтверждения операции. Перечислите и опишите основные параметры.

Практические задания:

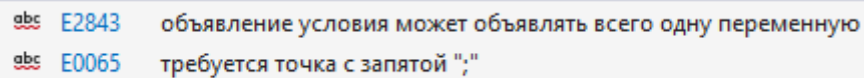
Необходимо проанализировать программу и устранить ошибки. Выполнить тестирование исправленной программы.

Задание № 1

```
#include <iostream>
using namespace std;
int main()
{
    for (int i, i < 10, i++)
    {
        cout << i << endl;
    }
}
```

Решение:

Были обнаружены следующие ошибки:

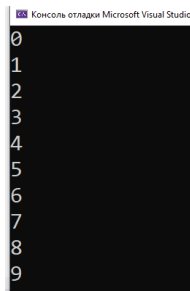


```
abc E2843 объявление условия может объявлять всего одну переменную
abc E0065 требуется точка с запятой ";"
```

В цикле for требуется указать вместо запятых точки с запятой, а также инициализировать переменную i:

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 0; i < 10; i++)
    {
        cout << i << endl;
    }
}
```

Выполним тестирование программы:



```
Консоль отладки Microsoft Visual Studio
0
1
2
3
4
5
6
7
8
9
```

Задание № 2

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 10, i < 10, i--)
    {
        cout << i << endl;
    }
}
```

Решение:

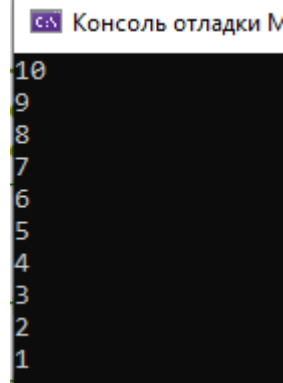
Были обнаружены следующие ошибки:

✗	C2086	int i: переопределение
!	Int-uninit	Локальная переменная не инициализирована.
✗	C2143	синтаксическая ошибка: отсутствие "," перед "<"
✗	C2143	синтаксическая ошибка: отсутствие ";" перед "{"
!		см. объявление "i"
abc	E0029	требуется выражение
abc	E0065	требуется точка с запятой ";"

В цикле for требуется указать вместо запятой точки с запятой, а также правильно указать условие в цикле:

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 10; i > 0; i--)
    {
        cout << i << endl;
    }
}
```

Выполним тестирование программы:



Задание № 3

```
#include <iostream>
using namespace std;
int main()
{
    for ( i = 10, i > 0, i--)
    {
        cout << i << endl;
    }
}
```

Решение:

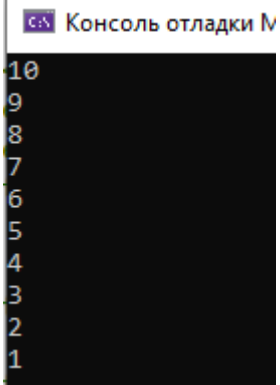
Были обнаружены следующие ошибки:

✗	C2065	i: необъявленный идентификатор
✗	C2065	i: необъявленный идентификатор
✗	C2065	i: необъявленный идентификатор
✗	C2065	i: необъявленный идентификатор
abc	E0020	идентификатор "i" не определен
✗	C2143	синтаксическая ошибка: отсутствие ")" перед "{"
✗	C2143	синтаксическая ошибка: отсутствие ";" перед ")"
✗	C2143	синтаксическая ошибка: отсутствие ";" перед "{"
abc	E0065	требуется точка с запятой ";"

В цикле for требуется указать вместо запятой точки с запятой, а также для переменной i написать тип данных int:

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 10; i > 0; i--)
    {
        cout << i << endl;
    }
}
```

Выполним тестирование программы:



Консоль отладки IV

```
10
9
8
7
6
5
4
3
2
1
```

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Ухтинский государственный технический университет»
(УГТУ)

филиал Ухтинского государственного технического университета
в г. Усинске
(УФ УГТУ)

(среднего профессионального образования)

УТВЕРЖДАЮ

И. о. директора филиала
О.В. Филиппова
« 26 » 20 26 г.



(подпись) (И. О. Фамилия)

« » 20 г.

(подпись) (И. О. Фамилия)

« » 20 г.

РАБОЧАЯ ПРОГРАММА

Дисциплина: Инструментальные средства разработки программного обеспечения

Индекс: МДК.02.02

Специальность: 09.02.11 Разработка и управление программным обеспечением

Форма обучения: очная

Курс (ы): 2

Семестр (ы): 3

г. Усинск

2026

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта среднего общего образования, утвержденного приказом Минпросвещения России от 24.02.2025 N 138 "Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением".

Разработчик Капембва Марине Андраниковна преподаватель СПО УФ УГТУ.

Рассмотрено на заседании ученого совета УФ УГТУ (протокол от 26.05.2026 № 06).

СОГЛАСОВАНО

Заместитель директора УФ УГТУ
по учебной работе



О.В. Филиппова