

Министерство образования и науки Челябинской области
Государственное бюджетное профессиональное образовательное учреждение
«Южно-Уральский государственный технический колледж»

**КОМПЛЕКТ
КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ**

по учебной дисциплине

«ОСНОВЫ ПРОГРАММИРОВАНИЯ»

программы подготовки специалистов среднего звена

по специальности 15.02.07
Автоматизация технологических процессов и
производств (по отраслям)
базовой подготовки

Челябинск, 2019 г.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств	4
1.1. Область применения комплекта контрольно-оценочных средств	4
1.2. Система контроля и оценки освоения программы учебной дисциплины	6
1.2.1. Формы промежуточной аттестации по учебной дисциплине	6
1.2.2. Организация текущего контроля успеваемости, промежуточной аттестации по итогам освоения учебной дисциплины	6
2. Задания для контроля и оценки освоения программы учебной дисциплины	7
2.1. Задания для текущего контроля	7
2.2. Задания для промежуточной аттестации	8
3. Литература	13

1. Паспорт комплекта контрольно-оценочных средств

1.1. Область применения комплекта контрольно-оценочных средств

Комплект контрольно-оценочных средств предназначен для контроля и оценки уровня освоения учебной дисциплины (далее УД) программы подготовки специалистов среднего звена (далее ППССЗ) по специальности СПО 15.02.07 Автоматизация технологических процессов и производств (по отраслям) (базовая подготовка).

Объектами контроля по УД являются элементы компетенций:

знания:

- этапы решения задач на компьютере
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования.

умения:

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;

Комплект контрольно-оценочных средств позволяет оценивать:

1) Формирование элементов профессиональных компетенций (ПК) и общих компетенций (ОК):

ПК 3.1. Выполнять работы по эксплуатации систем автоматического управления с учетом специфики технологического процесса.

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой

для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

2) Освоение умений и усвоение знаний

Таблица 1.

Освоенные умения, усвоенные знания	№№ заданий для проверки
У1. Работать в среде программирования;	ПР 1-10 СР 1-2
У2. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;	ПР 1-10 СР 1-2
З1. Этапы решения задач на компьютере;	ПР 1-10 СР 1-2
З2. Типы данных;	ПР 1-10 СР 1-2
З3. Базовые конструкции изучаемых языков программирования;	ПР 1-10 СР 1-2
З4. Принципы структурного и модульного программирования.	ПР 1-10 СР 1-2

Условные сокращения:

У – умения;

З – знания;

СР – самостоятельная работа (внеаудиторная);

ПР – практическая работа.

1.2 Система контроля и оценки освоения программы учебной дисциплины

1.2.1. Формы промежуточной аттестации по УД

Таблица 2

Форма промежуточной аттестации	Семестр
1	2
Дифференцированный зачёт	IV семестр

1.2.2. Организация текущего контроля успеваемости, промежуточной аттестации по итогам освоения программы учебной дисциплины

Оценка уровня освоения умений и усвоения знаний по учебной дисциплине производится по результатам практических занятий и выполнению заданий внеаудиторных самостоятельных работ.

Формой итоговой аттестации по учебной дисциплине является дифференцированный зачёт.

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся за работу, выполненную безошибочно, в полном объеме с учетом рациональности выбранных решений;
- оценка «хорошо» выставляется обучающемуся за работу, выполненную в полном объеме с недочетами;
- оценка «удовлетворительно» выставляется обучающемуся за работу, выполненную не в полном объеме (не менее 50% правильно выполненных заданий от общего объема работы);
- оценка «неудовлетворительно» выставляется обучающемуся за работу, выполненную не в полном объеме (менее 50% правильно выполненных заданий от общего объема работы).

2. Задания для контроля и оценки освоения программы учебной дисциплины

2.1 Задания для текущего контроля

2.1.1 Задания для оценки освоения умений

Перечень практических работ по дисциплине «Основы программирования»:

№	Наименование работы
1	Создание программ линейной структуры
2	Написание программ разветвляющейся структуры
3	Использование оператора выбора при написании программ
4	Разработка программ с использованием циклов
5	Создание программ циклической структуры
6	Создание пользовательских функций
7	Работа с одномерными массивами
8	Работа с двумерными массивами
9	Работа со строчными данными
10	Работа с текстовыми файлами средствами c++

Содержание заданий практических работ изложено в методических рекомендациях по выполнению практических работ по учебной дисциплине «Основы программирования».

2.1.2 Задания для оценки усвоения знаний и освоения умений

Перечень тем внеаудиторной самостоятельной работы

Наименование раздела	Вид заданий
Темы 1.1 – 2.7	выполнение индивидуальных заданий; написание и отладка программ различной сложности.

Содержание заданий самостоятельных работ изложено в методических рекомендациях по организации внеаудиторной самостоятельной работы по учебной дисциплине «Основы программирования».

2.2. Задания для промежуточной аттестации

Вопросы для подготовки к дифференцированному зачёту

1. Основы синтаксиса языка Си, структура консольного приложения.
2. Фундаментальные типы данных (bool, char, int, double).
3. Определение переменных и констант.
4. Оператор sizeof().
5. Выражения, операции, комментарии.
6. Оператор приведения типа.
7. Операторы инкремента и декремента.
8. Приоритет операторов в выражениях.
9. Блоки и правила видимости переменных.
10. Условный оператор и оператор перехода (if, goto).
11. Оператор множественной альтернативы (switch).
12. Цикл while. Прерывание цикла. Переход к следующей итерации.
13. Цикл do ... while. Прерывание цикла. Переход к следующей итерации.
14. Цикл for. Прерывание цикла. Переход к следующей итерации.
15. Математические функции стандартной библиотеки Си ().
16. Форматированный консольный ввод (): параметры функции scanf().
17. Форматированный консольный вывод (): параметры функции printf().
18. Форматированный файловый ввод-вывод ().
19. Бесформатный файловый ввод-вывод ().
20. Массивы. Передача массивов в параметрах функции.
21. Определение функции. Прототип функции. Рекурсия.
22. Параметры функции main()
23. Раздельная компиляция программных модулей. Использование *.h файлов.
24. Внешние (extern) и глобальные переменные.
25. Статические (static) переменные.
26. Статические (static) функции.
27. Указатели и операторы, с ними связанные.
28. Указатель на функцию.
29. Функции для работы с динамической памятью malloc()/realloc()/free().
30. Строки Си. Функции для работы со строками ().
31. Функции для работы с символами ().
32. Директивы препроцессора для условной компиляции и их использование.
33. Директивы препроцессора для включения файлов и их использование.

1. Вариант

1. Какие базовые типы данных Вы знаете? Сколько места они занимают в памяти компьютера?
2. Что означает команда «`continue`» и где она применяется?
3. Выделите память под вещественный массив чисел двойной точности из 123 элементов.

2. Вариант

1. Какие операторы языка Си Вы знаете? Какой у них приоритет?
2. Какие виды циклов Вы знаете? Чем они отличаются друг от друга?
3. Что такое прототип функции и где он используется?

3. Вариант

1. Что такое операции постфиксного инкремента и префиксного декремента?
2. Что означает команда «`break`» и где она применяется?
3. Что такое рекурсивная функция? Приведите пример.

4. Вариант

1. Что означает строка:

```
void func(int, int*);
```

2. Что такое метка и где она используется?
3. Что такое цикл «с предусловием» и цикл «с постусловием»?

5. Вариант

1. Сколько места в памяти компьютера занимают вещественные переменные и переменные с двойной точностью? Что такое точность представления вещественного числа?
2. Что означает команда «`goto`» и где она применяется?
3. Как найти заданную подстроку в другой строке?

6. Вариант

1. Что такое массив и что такое указатель? Приведите примеры.
2. Верно ли синтаксически написана программа:

```
void main( ) { }
```

3. Какие функции из библиотеки `<string.h>` Вы знаете?

7. Вариант

1. Что такое операция получения адреса? Где она используется?
2. Какие функции ввода-вывода Вы знаете?
3. Что такое перечислимый тип? Как он определяется?

8. Вариант

1. Как получить значение по указателю? Приведите примеры.
2. Какие форматирующие последовательности функции `printf()` Вы знаете?
3. Опишите структуру, состоящую из символьного массива из 12 членов, целого и вещественного числа.

9. Вариант

1. В каком случае можно изменить размер ранее выделенного массива? Как это сделать?
2. Напишите цикл `do ... while()`, меняющий порядок элементов массива на противоположный.
3. Для чего нужен оператор `typedef`?

10. Вариант

1. Напишите цикл `for`, меняющий порядок элементов массива на противоположный.
2. Как получить указатель на переменную?
3. Какие функции для работы с отдельными символами Вы знаете?

11. Вариант

1. Как записать массив в бинарный (неформатированный) файл?
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 1, b = 2, c = 3;  
c -= a += b = c;
```

3. Как вернуть из функции массив?

12. Вариант

1. Откройте файл `"myfile.txt"` на чтение.
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 1, b = 2;  
a += b++;
```

3. Что такое `union`?

13. Вариант

1. Как передать функцию в другую функцию? Как создать массив функций?
2. Как с помощью функции языка Си прочитать в три переменные три вещественных числа, разделенных пробелами?
3. Как обратиться к функции, определенной в другом файле с исходным текстом?

14. Вариант

1. Что такое заголовочный файл?
2. Как определить размер вещественного массива в байтах?
3. Напишите функцию, которая складывает два вещественных числа и возвращает результат.

15. Вариант

1. Для чего нужны статические переменные? Чем они отличаются от глобальных?
2. Что такое макроопределение с параметром? Приведите пример.
3. Как распечатать на экран строки из текстового файла, открытого на чтение?

16. Вариант

1. Как найти длину текстовой строки?
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 1, b = 4, c = 3;  
a = --b > c;
```

3. Для чего применяется оператор `default`?

17. Вариант

1. Как реализовать компиляцию некоторой функции в зависимости от условия?
2. Напишите циклы `while`, транспонирующие матрицу.
3. Опишите структуру, состоящую из имени, фамилии и года рождения студента.

18. Вариант

1. Определите двумерный вещественный массив.
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 1; int* b = &a;  
(*b)++;
```

3. Определите массив из трех структур, которые представляют разные точки на плоскости.

19. Вариант

1. Как изменить значение глобальной переменной, заданной в другом файле исходного текста проекта?
2. Какие форматирующие последовательности функции `printf()` Вы знаете?
3. Как задать перечислимый тип из трех убывающих элементов?

20. Вариант

1. Выделить динамическую память под строку из 20 символов.
2. Как передать аргументы с командной строки в функцию `main()`?
3. Распечатайте число π с тремя знаками после десятичной точки.

21. Вариант

1. Как закрыть функцию от использования в других исходных текстах проекта?
2. Какое значение будет иметь переменная `x`:

```
double x = 4;  
x *= 1/2;
```

3. Определите функцию, которая возвращает сумму элементов переданного массива.

22. Вариант

1. Как найти в строке любой из символов, входящих в строку "abc"?
2. Как напечатать целое число с обязательным выводом знака «плюс» для положительных чисел? Как сделать выравнивание по левой границе поля для этого числа?
3. Как принудительно преобразовать тип целого числа к вещественному типу?

23. Вариант

1. Как получить адрес третьего элемента вещественного массива? Если этот адрес уменьшить на единицу – куда он будет указывать?
2. Какое значение получит переменная **z** в результате выполнения программы:

```
int a = 1, b = 2, c = 3, d = 4; bool z;  
z = a < b || b < c && d < c;
```

3. Как определить, является ли заданный символ пробельным?

24. Вариант

1. Создайте макроопределение, которое вычисляет сумму двух своих аргументов.
2. Что делает функция **scanf ()** ?
3. Откройте файл "myfile.txt" на запись в режиме «запись в конец файла».

25. Вариант

1. Как проверить, открылся ли файл?
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 0, b = 1, c = 1;  
a = b << c++;
```

3. Как обратиться к переменной, определенной в ином исходном тексте проекта?

Критерии оценивания

- оценка «отлично» выставляется обучающемуся за работу, выполненную безошибочно, в полном объеме с учетом рациональности выбранных решений;
- оценка «хорошо» выставляется обучающемуся за работу, выполненную в полном объеме с недочетами;
- оценка «удовлетворительно» выставляется обучающемуся за работу, выполненную не в полном объеме (не менее 50 % правильно выполненных заданий от общего объема работы);
- оценка «неудовлетворительно» выставляется обучающемуся за работу, выполненную в не полном объеме (менее 50% правильно выполненных заданий от общего объема работы).

ЛИТЕРАТУРА

Основная литература:

1. Огнева М.В., Кудрина Е.В. Программирование на языке C⁺⁺: практический курс [текст]: учеб. пособие для СПО/ М.В.Огнева, Е.В.Кудрина. - М.: Издательство Юрайт, 2019 г. – 335 с.

Дополнительная литература:

2. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C⁺⁺ [электронный ресурс]: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва: ИД «ФОРУМ»: ИНФРА-М, 2018. — 512 с. — (Среднее профессиональное образование). - Режим доступа: <https://znanium.com/catalog/product/918098>
3. Воронцова, Е. А. Программирование на C⁺⁺ с погружением: практические задания и примеры кода [электронный ресурс]. - Москва: НИЦ ИНФРА-М, 2016. - 80 с.- Режим доступа: <https://znanium.com/catalog/product/563294>

Интернет - ресурсы

1. www.ict.equ.ru/catalog - ИК Портал - Интернет – ресурсы;
2. www.intuit.ru/departament - Интернет университет - информационных технологий.