

Министерство науки и высшего образования Российской Федерации
Лысьвенский филиал федерального государственного автономного образовательного
учреждения высшего образования
«Пермский национальный исследовательский политехнический университет

УТВЕРЖДАЮ

Доцент с исп. обяз. завкафедрой ТД

 Т.О. Сошина

«30» августа 2021 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

**для проведения текущего контроля успеваемости и промежуточной
аттестации обучающихся по дисциплине**

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

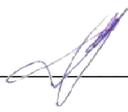
основной профессиональной образовательной программы
подготовки специалистов среднего звена

по специальности СПО 09.02.07 Информационные системы и программирование

Лысьва, 2021

Фонд оценочных средств разработан на основе:

- Федерального государственного образовательного стандарта среднего профессионального образования, утверждённого приказом Министерства образования и науки Российской Федерации «9» декабря 2016 г. № 1547 по специальности 09.02.07 Информационные системы и программирование;
- рабочей программы учебной дисциплины Основы алгоритмизации и программирования, утвержденной «30» августа 2021г.

Разработчик: преподаватель высшей категории _____  Федосеева Е.Л

Фонд оценочных средств рассмотрен и одобрен на заседании предметной (цикловой) комиссии естественнонаучных дисциплин (ПЦК ЕНД) «30» августа 2021 г., протокол № 01.

Председатель ПЦК ЕНД



М.Н. Апталаев

ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения учебной дисциплины Основы алгоритмизации и программирования обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование следующими результатами обучения: знаниями и умениями, которые формируют общие компетенции¹.

Показатели, критерии, средства оценивания достижения запланированных результатов обучения и шкала оценки результатов формирования частей компетенций, проверяемых в при текущем и промежуточном контроле представлены в таблице 1.

Формой промежуточной аттестации по учебной дисциплине является в первом семестре **дифференцированный зачёт** и во втором семестре **экзамен**.

КОНТРОЛЬ РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

1. ТЕКУЩИЙ И ПРОМЕЖУТОЧНЫЙ КОНТРОЛЬ ОСВОЕНИЯ ЗАДАНЫХ ДИСЦИПЛИНАРНЫХ КОМПЕТЕНЦИЙ

Текущий и промежуточный контроль освоения дисциплинарных компетенций проводится в следующих формах:

- устный опрос,
- тестирование,
- отчеты по практическим и лабораторным занятиям.

Уровень освоения частей компетенций подтверждается оценкой по четырехбалльной шкале во время текущего контроля успеваемости, определяемой исходя из количества средние набранных баллов по каждому результату обучения по дисциплине, в соответствии с показателями, критериями и шкалой оценивания, представленными в таблице 1.

¹ Введены в действие новые формулировки общих компетенций с 22.10.2022 г. на основании приказа Минпросвещения России от 01.09.2022 № 796 «О внесении изменений в ФГОС СПО»

Таблица 1 - Показатели, критерии, средства оценивания достижения запланированных результатов обучения и шкала оценки результатов формирования частей компетенций, приобретаемых в ходе освоения дисциплины Основы алгоритмизации и программирования

Результаты обучения	Показатели и критерии оценивания сформированности частей компетенций		Средства оценивания	Шкала оценивания		
	показатели	критерии		5	4	3
<p>ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 1.1-1.5, ПК 2.4-2.5, ЛР 16-19, ЛР 22, ЛР 23, ЛР 25, ЛР 26, ЛР 28</p> <p>Уметь:</p> <ul style="list-style-type: none"> – разрабатывать алгоритмы для конкретных задач; – использовать программы для графического отображения алгоритмов; – определять сложность работы алгоритмов; – работать в среде программирования; – реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; – оформлять код программы в соответствии со стандартом кодирования; – выполнять проверку, отладку кода программы 	<p>Правильно выполненное и обоснованное решение задач</p> <p>Правильность выбора алгоритма решения задач, правильность разработанного алгоритма задачи</p> <p>Качество оформления полученных результатов и кода программы</p> <p>Оформление отчета по практическим и лабораторным занятиям</p>	<p>Правильность выбора методы и алгоритма решения задач, корректность работы алгоритма и программы, верность сформулированных выводов</p> <p>Объективность и достоверность полученных данных</p> <p>Во время сданные и правильно оформленные отчёты по практическим и лабораторным занятиям</p>	<p>Практические занятия 1-15, Лабораторные занятия 1-14</p>	<p>Правильно выполнены все задания , даны ответы на все контрольные вопросы, своевременно предоставлен отчёт о выполненной работе</p>	<p>Правильно выполнены все задания, правильно даны ответы на большую часть контрольных вопросов, несвоевременно предоставлен отчёт о выполненной работе, либо в случае своевременного предоставления отчёта, но с наличием несущественных ошибок в выполнении заданий и/или ответах на контрольные вопросы, не противоречащие основным понятиям дисциплины</p>	<p>Выполнены не все, но более 50% заданий работы, дан ответ на часть контрольных вопросов, имеются несущественные ошибки в выполнении заданий и/или ответах на контрольные вопросы, не противоречащие основным понятиям дисциплины</p>
<p>ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 1.1-1.5, ПК 2.4-2.5, ЛР 16-19, ЛР 22, ЛР 23, ЛР 25, ЛР 26, ЛР 28</p> <p>Знать:</p>	<p>Понимание сути основных понятий алгоритмизации, принципов построения алгоритмов, основ конструкций языка программирования и его</p>	<p>Количество правильных ответов в тесте на знание основных понятий алгоритмизации, принципов построения</p>	<p>Тесты по разделам</p>	<p>90-100%</p>	<p>76-90%</p>	<p>51-75%</p>

<ul style="list-style-type: none"> – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; – эволюцию языков программирования, их классификацию, понятие системы программирования; – основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти; – понятие подпрограммы, принципы составления библиотек подпрограмм; – объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения 	<p>частей</p>	<p>алгоритмов, основ конструкций языка программирования и его частей</p>				
		<p>Знание материала Последовательность изложения Владение речью и терминологией. Применение конкретных примеров</p>	<p>Устный ответ по всем разделам</p>	<p>Точное, уверенное воспроизведение материала</p>	<p>Достаточно точное воспроизведение</p>	<p>Допущены отдельные ошибки, и неточности</p>

ТИПОВЫЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

1. Типовые вопросы для устного опроса

Критерии и шкалы оценивания представлены в таблице 1.

Вопросы для устного опроса

Раздел «Введение в программирование»

1. Перечислите области применения языков программирования
2. Назовите языки программирования высокого уровня
3. Перечислите основные этапы решения задач на компьютере
4. Что такое компилятор?
5. Что такое интерпретатор?
6. Назовите основные типы переменных языка C#
7. Опишите неявное преобразование типа переменных, приведите примеры
8. Опишите явное преобразование типа переменных, приведите примеры

Раздел «Основные конструкции языков программирования»

1. Опишите структуру программы на алгоритмическом языке
2. Опишите лексику языка C#
3. Как описываются константы на языке C#
4. Перечислите основные операторы языка C#
5. Как записываются выражения на языке C#?
6. Что такое составной оператор?
7. Что такое оператор выбора?
8. Как записываются вложенные условные операторы?
9. Какие виды циклов существуют?
10. Опишите цикл с постусловием
11. Опишите цикл с предусловием
12. Опишите цикл с параметром
13. Перечислите основные операторы перехода
14. Что такое вложенные циклы?
15. Что такое массив?
16. Как объявляется массив на языке программирования C#?
17. Перечислите стандартные функции для массива целых чисел
18. Перечислите стандартные функции для массива вещественных чисел
19. Перечислите стандартные функции для работы со строками
20. Как происходит объявление множества
21. Перечислите операции над множествами

Раздел «Структурное программирование»

1. Что такое подпрограмма?
2. Для чего нужны подпрограммы?
3. В чем заключается различие между процедурами и функциями?
4. Что такое рекурсивная функция?
5. Что такое формальные параметры?
6. Что такое фактические параметры?
7. С какими типами файлов приходится работать в программировании?
8. Как организуется доступ к файлам?
9. Какие основные операторы используются при работе с файлами
10. Какие существуют методы структурного программирования
11. В чем заключается основная разработка программы в структурном программировании
12. Что такое модуль?
13. Опишите структуру модуля
14. Назовите стандартные модули в C#
15. В чем заключается компиляция и компоновка программы

16. Что такое указатели?
17. Что такое динамическое распределение памяти?
18. Как создаются динамические переменные?
19. Как удаляются динамические переменные?
20. Опишите структуру данных на основе указателей

Раздел «Введение в объектно-ориентированное программирование»

1. Что такое объект?
2. Что такое объектно-ориентированное программирование?
3. Что такое инкапсуляция?
4. Что такое наследование?
5. Что такое полиморфизм?
6. Как осуществляется доступ к элементам класса?
7. Как осуществляется доступ к элементам наследования?
8. Что такое виртуальные методы?
9. Как происходит динамическое связывание методов
10. Перечислите основные этапы реализации полиморфизма
11. Какие существуют абстрактные методы в полиморфизме?
12. Что такое классы объектов?
13. Что такое компоненты и их свойства?
14. Какие предъявляются требования к аппаратным и программным средствам интегрированной среды разработчика?
15. Опишите состав проекта
16. Как происходит настройка среды параметров проекта?
17. Перечислите основные компоненты интегрированной среды разработки
18. Что такое хеширование?
19. Какие существуют особенности при разработке функционального интерфейса приложения?
20. С чего начинается создание интерфейса пользователя?
21. Что такое тестирование приложения?
22. Что такое перегрузка методов?
23. Что такое сериализация?
24. Как происходит сериализация объектов?
25. Как происходит десериализация объектов?
26. С помощью чего происходит организация взаимодействия объектов?

2. Типовые тесты по разделам

Критерии и шкалы оценивания представлены в таблице 1.

Типовой тест № 1

Раздел «Введение в программирование», Раздел «Основные конструкции языков программирования»

Условия выполнения задания

- тест выполняется в аудитории во время практических занятий;
- для выполнения теста необходимо следующее оборудование: бланки ответов, ручки, карточки с тестами (для выполнения электронного варианта теста: компьютерный класс, тестировщик).

Инструкция: на выполнение теста отводится 20 минут, внимательно прочитайте вопрос, выберите один вариант ответа, ответы занесите в бланк ответов

1. Установите соответствие

bool		'g'
int		"Hello"
double		45

- произвольная группа команд?
10. Что называется массивом?
- 1) Под *массивом* понимается совокупность конечного числа данных различных типов.
 - 2) Под *массивом* понимается совокупность конечного числа данных одного типа.
 - 3) Под *массивом* понимается совокупность бесконечного числа данных одного типа

Типовой тест № 2

Раздел «Структурное программирование»

Условия выполнения задания

- тест выполняется в аудитории во время практических занятий;

- для выполнения теста необходимо следующее оборудование: бланки ответов, ручки, карточки с тестами (для выполнения электронного варианта теста: компьютерный класс, тестировщик).

Инструкция: на выполнение теста отводится 15 минут, внимательно прочитайте вопрос, выберите один вариант ответа, ответы занесите в бланк ответов

1. Когда аргумент передается в функцию по ссылке ...
 - а) внутри функции создается переменная, хранящая значение этого аргумента
 - б) функция не имеет доступа к значению аргумента
 - в) в вызывающей программе создается временная переменная для хранения значения аргумента
 - г) функция получает доступ к параметру в вызывающей программе
2. Когда аргумент передается в функцию по значению
 - А) внутри функции создается переменная, хранящая значение этого аргумента
 - Б) функция не имеет доступа к значению аргумента
 - в) в вызывающей программе создается временная переменная для хранения значения аргумента
 - г) функция получает доступ к параметру в вызывающей программе
3. Переменная, описанная внутри блока, видима
 - 1) от точки своего объявления до конца программы
 - 2) от точки своего объявления до конца функции
 - 3) только внутри функции
 - 4) от точки от точки своего объявления до конца блока
4. Какое число будет выведено в результате работы программы?

```
#include <stdio.h>
#include <math.h>
int main()
{
const int a[8] = {3, 8, 0, -6, 0, -1, -9, 3};
int i, k;
k = abs(a[7]);
for (i = 0; i < 8; i++)
{
if (abs(a[i]) > k) k = abs(a[i]);
printf("%d", k);
}
```

```
    }  
}
```

Ответ: _____

5. Что будет выведено на экран в результате работы программы?

```
#include <stdio.h>  
const char a[] = "abcra";  
int i; char k;  
main()  
{  
    k = 'c';  
    for (i = 1; i < 5; i++)  
        if (a[i] < k) {  
            k = a[i];  
            printf("%c", k);  
        }  
}
```

Ответ: _____

6. Выберите пример указателя на константу

- А) char* const pc
- Б) char* pc
- В) const char* const pc
- Г) невозможно определить указатель на константу.
- Д) const char* pc

Типовой тест № 3

Раздел «Введение в объектно-ориентированное программирование»

Условия выполнения задания

- тест выполняется в аудитории во время практических занятий;

- для выполнения теста необходимо следующее оборудование: бланки ответов, ручки, карточки с тестами (для выполнения электронного варианта теста: компьютерный класс, тестировщик).

Инструкция: на выполнение теста отводится 15 минут, внимательно прочитайте вопрос, выберите один вариант ответа, ответы занесите в бланк ответов

1. Свойство объектов, при котором действие с одинаковыми именами вызывает различное поведение для различных объектов
 - А) Полиморфизм
 - Б) Передача
 - В) Наследование
 - Г) Монорфизм
2. Данные, характеризующие состояние объекта
 - а) Доли объекта
 - б) Части объекта
 - в) Атрибуты объекта
3. Совокупность объектов, характеризующаяся общностью методов и свойств
 - а) Класс
 - б) Вид
 - в) Род

4. Действие, которое может выполнить объект
 - а) Метод
 - б) Событие
 - в) Свойство
5. Совокупность свойств и методов
 - а) Объект;
 - б) Свойство;
 - в) Событие.
6. Компоненты, которые видны во время работы приложения, с ними напрямую может взаимодействовать пользователь, называются ...
 - а) Абстрактными
 - б) Видимыми
 - в) Визуальными
7. Какая из перечисленных функций не может быть конструктором
 - а) String (const int a)
 - б) String (String & s)
 - в) void String ()
8. Свойство Name отвечает за ...
 - а) Название компонента
 - б) Имя компонента
 - в) Назначение компонента
9. Этот компонент предназначен для вывода текста на поверхность формы.
 - А) Label
 - Б) Edit
 - В) Button
10. Скрытие деталей реализации называется ...
 - А) Инкапсуляцией
 - Б) Наследование
 - В) Полиморфизм
 - Г) Абстракцией

3. Практические и лабораторные занятия

Критерии и шкалы оценивания представлены в таблице 1.

Список практических занятий:

1. Знакомство со средой программирования
2. Составление программ линейной структуры
3. Составление программ разветвляющей структуры
4. Составление программ циклической структуры
5. Обработка одномерных массивов. Обработка двумерных массивов
6. Работа со строками
7. Работа с данными типа множество
8. Организация процедур. Организация функций
9. Применение рекурсивных функций
10. Файлы последовательного доступа. Типизированные файлы
11. Нетипизированные файлы

12. Создание библиотек подпрограмм
13. Классы ООП: виды, назначение, свойства, методы, события. Объявление класса
14. Создание наследованного класса
15. Изучение интегрированной среды разработчика

Список лабораторных занятий:

1. Программирование модуля
2. Использование указателей для организации связанных списков
3. События компонентов (элементы управления), их сущность и назначение
4. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени
5. Создание проекта с использованием компонентов для работы с текстом
6. Создание процедур на основе событий
7. Создание проекта с использованием кнопочных компонентов
8. Создание проекта с использованием компонентов стандартных диалогов и системы меню
9. Разработка оконного приложения с несколькими формами
10. Создание игрового приложения
11. Разработка интерфейса пользователя
12. Тестирование, отладка приложения
13. Перегрузка методов
14. Программирование приложений

2. ИТОГОВЫЙ КОНТРОЛЬ ОСВОЕНИЯ ЗАДАНЫХ ДИСЦИПЛИНАРНЫХ КОМПЕТЕНЦИЙ

Итоговый контроль освоения заданных дисциплинарных компетенций проводится во время промежуточной аттестации в форме дифференцированного зачёта в первом семестре и экзамена во втором семестре

Условия проставления дифференцированного зачёта в первом семестре по дисциплине: дифференцированный зачёт по дисциплине «Основы алгоритмизации и программирования» выставляется по итогам проведённого текущего контроля знаний студентов и выставленной средней результирующей оценки по всем модулям текущего контроля:

- оценка «отлично» за дисциплину – средняя оценка по всем модулям не менее 4,5;
- оценка «хорошо» за дисциплину – средняя оценка по всем модулям не менее 4,0;
- оценка «удовлетворительно» за дисциплину – средняя оценка по всем модулям не менее 3,0

Экзамен по дисциплине проводится по билетам. Билет содержит два теоретических вопроса и одно практическое задание. К сдаче экзамена допускаются студенты, сдавшие выполненные задания по практическим и лабораторным заданиям и получившие оценки не ниже «удовлетворительно» по результатам текущего контроля успеваемости. Итоговая экзаменационная оценка выставляется с учётом результатов текущего контроля успеваемости, если средняя оценка по результатам текущего контроля успеваемости выше 4,5, то освобождаются от одного теоретического вопроса по выбору студента. Итоговая оценка по дисциплине выставляется как взвешенная сумма экзаменационной оценки и результирующих оценок за все модули прохождения дисциплины (результатов текущего контроля успеваемости):

$$O_{\text{итоговая}} = 0,6 * O_{\text{ср.результат}} + 0,4 * O_{\text{экз.}}$$

Уровень освоения частей компетенций подтверждается оценкой по дисциплине, определяемой исходя из количества средне набранных баллов по каждому контрольному заданию билета, в соответствии с показателями, критериями и шкалой оценивания, представленными в таблице 2.

Типовые вопросы для подготовки к дифференцированному зачету

Типовые вопросы для контроля усвоенных знаний:

- 1 Развитие языков программирования. Обзор языков программирования.
- 2 Области применения языков программирования.
- 3 Стандарты языков программирования.
- 4 Среда проектирования.
- 5 Компиляторы и интерпретаторы.
- 6 Жизненный цикл программы
- 7 Программа. Программный продукт и его характеристики.
- 8 Основные этапы решения задач на компьютере
- 9 Переменные: объявление переменных, именованное пространство имен.
- 10 Система типов языка C#. Встроенные типы. Типы CTS.
- 11 Преобразование типов: неявное преобразование типа, явное преобразование типа, операции преобразования для данных строкового типа.
- 12 Консольный ввод и вывод

- 13 Структурная схема программы на алгоритмическом языке
- 14 Лексика языка C#. Переменные и конспекты. Выражения и операции.
- 15 Операторы языка C#: операторы объявления, операторы выражения.
- 16 Синтаксис операторов C#
- 17 Составной оператор. Вложенные условные операторы. Оператор выбора.
- 18 Применение алгебры логики при создании алгоритмов
- 19 Цикл с постусловием.
- 20 Цикл с предусловием.
- 21 Цикл с параметром
- 22 Вложенные циклы. Операторы перехода: break, continue, goto, return
- 23 Структуры данных. Массивы. Объявление массива. Ввод и вывод одномерных и двумерных массивов.
- 24 Обработка массивов. Стандартные функции для массива целых и вещественных чисел
- 25 Простейшие алгоритмы поиска в массиве.
- 26 Алгоритмы сортировки в массиве
- 27 Строки и множества. Объявление строковых типов данных.
- 28 Операции со строками. Стандартные функции и процедуры работы со строками
- 29 Объявление множества. Операции над множествами

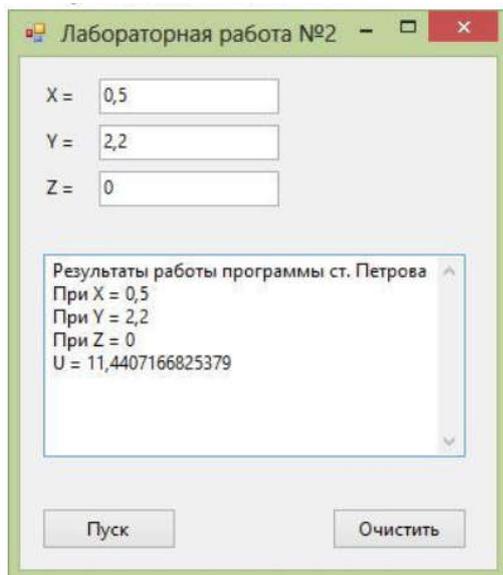
Типовые задания для контроля освоенных умений

1. Разместите на форме четыре кнопки (Button). Сделайте на кнопках следующие надписи: «красный», «зеленый», «синий», «желтый». Создайте четыре обработчика события нажатия на данные кнопки, которые будут менять цвет формы в соответствии с текстом на кнопках.
2. Составить программу вычисления для заданных значений x , y , z арифметического выражения:

$$u = tg^2(x + y) e^{y-z} \sqrt{\cos x^2 + \sin z^2}$$

3. Создать форму, которая выполняет вычисления для функции

$$u = \begin{cases} y \sin(x) + z, z & x = 0 \\ ye^{\sin(x)} & z, z & x < 0 \\ y \sin(\sin(x)) + z, z & x > 0 \end{cases}$$



4. Вычислить и вывести на экран таблицу значений функции $y = a \cdot \ln(x)$ при x , изменяющемся от x_0 до x_k с шагом dx , a – константа
5. В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом
6. Дана матрица $A(3,3)$. Вычислить сумму второй строки и произведение первого столбца. Вывести исходную матрицу и результаты вычислений
7. Написать программу подсчета числа слов в произвольной строке. В качестве разделителя может быть любое число пробелов. Для ввода строк использовать ListVox. Строки вводятся на этапе проектирования формы, используя окно свойств. Вывод результата организовать в метку Label

Типовые вопросы для подготовки к экзамену

Типовые вопросы для контроля усвоенных знаний:

- 1 Понятие подпрограммы. Процедуры и функции, их сущность, назначение и различие.
- 2 Организация процедур, стандартные процедуры.
- 3 Процедуры, определенные пользователем: синтаксис, передача аргументов.
- 4 Формальные и фактические параметры.
- 5 Процедуры с параметрами, описание процедур.
- 6 Рекурсивные функции.
- 7 Типы файлов. Организация доступа к файлам.
- 8 Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа.
- 9 Файлы произвольного доступа. Порядок работы с файлами произвольного доступа
- 10 Основы структурного программирования. Методы структурного программирования.
- 11 Стили структурного программирования. Разработка программы
- 12 Понятие модуля. Структура модуля.
- 13 Компиляция и компоновка программы.
- 14 Стандартные модули
- 15 Описание указателей. Основные понятия и применение динамически распределяемой памяти.
- 16 Создание и удаление динамических переменных. Структуры данных на основе указателей.
- 17 Задача о стеке
- 18 История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс
- 19 Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
- 20 Понятие наследования. Особенности базового и производственного классов.
- 21 Доступ к элементам класса и наследование
- 22 Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Понятие полиморфизма.
- 23 Ссылка на объект базового класса и объекты производственных классов. Использование ссылки на объект базового класса
- 24 Виртуальные методы. Динамическое связывание.
- 25 Основные этапы реализации полиморфизма.
- 26 Приведение типов объектов. Абстрактные методы
- 27 Классы объектов. Компоненты и их свойства.
- 28 Событийно-управляемая модель программирования.

- 29 Компонентно-ориентированный подход
- 30 Требования к аппаратным и программным средствам интегрированной среды разработчика
- 31 Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты.
- 32 Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта
- 33 Состав и характеристика проекта.
- 34 Выполнение проекта. Настройка среды и параметров проекта
- 35 Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
- 36 Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат
- 37 Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение
- 38 Создание процедур на основе событий
- 39 Диалоги. Класс диалогов. Формы. Хеширование
- 40 Разработка функционального интерфейса приложения.
- 41 Создание интерфейса приложения
- 42 Разработка функциональной схемы работы приложения.
- 43 Разработка игрового приложения
- 44 Разработка приложения. Проектирование объектно-ориентированного приложения.
- 45 Создание интерфейса пользователя.
- 46 Тестирование, отладка приложения
- 47 Классы ООП: виды, назначение, свойства, методы, события.
- 48 Перегрузка методов.
- 49 Создание многоуровневой иерархии. Абстрактные классы
- 50 Понятие сериализации. Сериализация объектов в бинарный файл.
- 51 Десериализация объектов из бинарного файла
- 52 Организация взаимодействия объектов. Реализация взаимодействия объектов

Типовые задания для контроля освоенных умений

1. Составить программу с тремя функциями, выводящих на экран символы #, @, \$
2. Составить программу для вычисления площади кольца по значениям внутреннего и внешнего радиусов, используя функцию вычисления площади круга. Функции передается один параметр – радиус и возвращается результат - вычисленная площадь
3. Напишите рекурсивную функцию, которая переворачивает элементы массива и обратном порядке
4. На основе базового класса «Автомобиль», создать произвольный класс «Общественный транспорт»
5. Создать функции, определяющие максимальное и минимальное значение переменных каждого объекта. Используйте конструктор
6. Разработать приложение «Калькулятор»
7. Расположить на форме четыре панели для отображения текущего времени, даты, названия дня недели и панель с секундомером. Подобрать цвета, стиль и расположение панелей на форме

8. Разработать приложение «Конвертор величин измерения» с использованием компонента ввода строки Edit. Создать программный код необходимых событий

Таблица 2 - Показатели, критерии, средства оценивания результатов обучения при промежуточной аттестации и шкала оценки результатов формирования частей компетенций, приобретаемых в ходе освоения дисциплины Основы алгоритмизации и программирования

Результаты обучения	Показатели и критерии оценивания сформированности частей компетенций		Средства оценивания	Шкала оценивания		
	показатели	критерии		5	4	3
<p>ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1-1.5, ПК 2.4-2.5, ЛР 16-19, ЛР 22, ЛР 23, ЛР 25, ЛР 26, ЛР 28</p> <p>Уметь:</p> <ul style="list-style-type: none"> – разрабатывать алгоритмы для конкретных задач; – использовать программы для графического отображения алгоритмов; – определять сложность работы алгоритмов; – работать в среде программирования; – реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; – оформлять код программы в соответствии со стандартом кодирования; – выполнять проверку, отладку кода программы 	<p>Понимание сути основ алгоритмов, использование программ для графического отображения алгоритмов, определения сложности работы алгоритмов, работы в среде программирования</p>	<p>Точность воспроизведения формулировок алгоритмов, использование программ для графического отображения алгоритмов, определения сложности работы алгоритмов, работы в среде программирования</p>	<p>Устный ответ на экзамене</p>	<p>Точное, уверенное воспроизведение содержания основ алгоритмов и программирования</p>	<p>Достаточно точное воспроизведение содержания алгоритмов и программирования</p>	<p>Допущены отдельные ошибки, и неточности в ответе</p>
	<p>Правильно выполненное и обоснованное решение задач</p>	<p>Объективность и достоверность полученных данных</p> <p>Правильность выбора методы и алгоритма решения задач, корректность проведенных расчетов, верность сформулированных выводов</p>	<p>Практические задания на экзамене</p>	<p>Глубокое исчерпывающее решение задач</p>	<p>Достаточно полное решение задач, при несущественных неточностях</p>	<p>Понимание алгоритма решения задач</p>
<p>ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ОК 10, ПК 1.1-1.5, ПК 2.4-2.5, ЛР 16-19, ЛР 22, ЛР 23, ЛР 25, ЛР 26, ЛР 28</p> <p>Знать:</p> <ul style="list-style-type: none"> – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; – эволюцию языков 	<p>Понимание сути алгоритмизации, эволюции языков программирования, элементов языка программирования, объектно-ориентированного программирования</p>	<p>Знание основных понятий алгоритмизации, эволюции языков программирования, элементов языка программирования, объектно-ориентированного программирования</p>	<p>Устный ответ на экзамене</p>	<p>Точное, уверенное воспроизведение понятий алгоритмизации, эволюции языков программирования, элементов языка программирования, объектно-ориентированного программирования комплексных чисел на практике</p>	<p>Достаточно точное воспроизведение понятий алгоритмизации, эволюции языков программирования, элементов языка программирования, объектно-ориентированного программирования</p>	<p>Допущены отдельные ошибки, и неточности при воспроизведении понятий алгоритмизации, эволюции языков программирования, элементов языка программирования, объектно-ориентированного программирования</p>

<p>программирования, их классификацию, понятие системы программирования;</p> <ul style="list-style-type: none"> – основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти; – понятие подпрограммы, принципы составления библиотек подпрограмм; – объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения 	<p>Правильно выполненное и обоснованное решение задач</p>	<p>Объективность и достоверность полученных данных</p> <p>Правильность выбора методы и алгоритма решения задач, корректность проведенных расчетов, верность сформулированных выводов</p>	<p>Практические задания на экзамене</p>	<p>Глубокое исчерпывающее решение задач</p>	<p>Достаточно полное решение задач, при несущественных неточностях</p>	<p>Понимание алгоритма решения задач</p>
---	---	--	---	---	--	--

Лист регистрации изменений

№ п.п.	Содержание изменения	Дата, номер протокола заседания ПЦК. Подпись председателя ПЦК
1	В соответствии с принятыми поправками к Федеральному закону № 273-ФЗ «Об образовании в Российской Федерации» с 01.09.2021 г. введены личностные результаты обучения.	30 августа 2021 № 01 Председатель ПЦК ЕНД  / М.Н. Апталаев
2	На основании Приказа Минпросвещения России от 01.09.2022 № 796 "О внесении изменений в ФГОС СПО" введены в действие новые формулировки общих компетенций с 22.10.2022 г. На основании внесенных изменений заменена страница 3, 4	11 октября 2022 № 02 Председатель ПЦК ЕНД  / М.Н. Апталаев