



Федеральное агентство морского и речного транспорта
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Государственный университет морского и речного флота
имени адмирала С.О. Макарова»**
Воронежский филиал ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»

Кафедра математики, информационных систем и технологий

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Языки программирования»
(приложение к рабочей программе дисциплины)

Направление подготовки 09.03.02 Информационные системы и технологии

Направленность (профиль) Информационные системы на транспорте

Уровень высшего образования бакалавриат

Форма обучения очная, заочная

Воронеж
2025

1. Перечень компетенций и этапы их формирования в процессе освоения дисциплины

В результате освоения ОПОП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

Таблица 1

Перечень компетенций и этапы их формирования в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции	Планируемые результаты обучения по дисциплине
ОПК-6: Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ОПК-6.1 Разработка алгоритмов, пригодных для практического использования	Знать: основы современных языков программирования; основы структурного программирования; классификацию языков программирования Уметь: формализовать поставленную задачу; применять соответствующий язык программирования при решении конкретных научных и практических задач Владеть: навыками использования современных технологий программирования, навыками разработки информационных систем для решения прикладных задач.
	ОПК-6.2 Разработка компьютерных программ, пригодных для практического использования	Знать: языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий. Уметь: применять языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий. Владеть: навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.

2. Паспорт фонда оценочных средств для проведения текущей и промежуточной аттестации обучающихся

Таблица 2

Оценочные средства для проведения текущего контроля и промежуточной аттестации обучающихся

№ п/п	Наименование раздела (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Тема 1. История, развитие и становление языков программирования	ОПК-6	тестирование, зачет
2	Тема 2. Объектно-ориентированное программирование	ОПК-6	тестирование, КР, зачет
3	Тема 3. Формальные грамматики и языки.	ОПК-6	тестирование, зачет
4	Тема 4. Проблема распознавания языка	ОПК-6	тестирование, зачет

Таблица 3

Критерии оценивания результата обучения по дисциплине и шкала оценивания по дисциплине

Результат обучения по дисциплине	Критерии оценивания результата обучения по дисциплине и шкала оценивания по дисциплине				Процедура оценивания
	2	3	4	5	
	Не зачтено	Зачтено			
ОПК-6.1 Знать: основы современных языков программирования ; основы структурного программирования ; классификацию языков программирования	Отсутствие или фрагментарные представления об основах современных языков программирования; основах структурного программирования; классификации языков программирования	Неполные представления об основах современных языков программирования; основах структурного программирования; классификации языков программирования	Сформированные, но содержащие отдельные пробелы представления об основах современных языков программирования; основах структурного программирования; классификации языков программирования	Сформированные систематические представления об основах современных языков программирования; основах структурного программирования; классификации языков программирования	Тестирование, зачет
ОПК-6.1 Уметь: формализовать поставленную задачу; применять соответствующий язык программирования при решении конкретных научных и практических задач	Отсутствие умений или фрагментарные умения формализовать поставленную задачу; применять соответствующий язык программирования при решении конкретных научных и практических задач	В целом удовлетворительные, но не систематизированные умения формализовать поставленную задачу; применять соответствующий язык программирования при решении конкретных научных и практических задач	В целом удовлетворительные, но содержащие отдельные пробелы умения формализовать поставленную задачу; применять соответствующий язык программирования при решении конкретных научных и практических задач	Сформированные умения формализовать поставленную задачу; применять соответствующий язык программирования при решении конкретных научных и практических задач	Тестирование, зачет
ОПК-6.1 Владеть: навыками использования современных технологий программирования, навыками разработки информационных систем для решения прикладных задач;	Отсутствие владения или Фрагментарное владение навыками использования современных технологий программирования, навыками разработки информационных систем для решения прикладных задач;	В целом удовлетворительные, но не систематизированные навыки использования современных технологий программирования, навыками разработки информационных систем для решения прикладных задач;	В целом удовлетворительные, но содержащие отдельные пробелы навыки использования современных технологий программирования, навыками разработки информационных систем для решения прикладных задач;	Сформированное владение навыками использования современных технологий программирования, навыками разработки информационных систем для решения прикладных задач;	Тестирование, КР, зачет

<i>ОПК-6.2</i> Знать: языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий	<i>Отсутствие или фрагментарные представления о языках и технологиях программирования, пригодных для практического применения в области информационных систем и технологий</i>	<i>Неполные представления о языках и технологиях программирования, пригодных для практического применения в области информационных систем и технологий</i>	<i>Сформированные, но содержащие отдельные пробелы представления о языках и технологиях программирования, пригодных для практического применения в области информационных систем и технологий</i>	<i>Сформированные систематические представления о языках и технологиях программирования, пригодных для практического применения в области информационных систем и технологий</i>	<i>Тестирование, зачет</i>
<i>ОПК-6.2</i> Уметь: применять языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий	<i>Отсутствие умений или фрагментарные умения применять языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий</i>	<i>В целом удовлетворительные, но не систематизированные умения применять языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы умения применять языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий</i>	<i>Сформированные умения применять языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий</i>	<i>Тестирование, зачет</i>
<i>ОПК-6.2</i> Владеть: навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач	<i>Отсутствие владения или Фрагментарное владение навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач</i>	<i>В целом удовлетворительные, но не систематизированные навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач</i>	<i>В целом удовлетворительные, но содержащие отдельные пробелы навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач</i>	<i>Сформированное владение навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач</i>	<i>Тестирование, КР, зачет</i>

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

Тест для проведения текущего контроля

1. Языки программирования:
 - а) аспектно-ориентированные
 - б) аспектно-направленные
 - в) аспектно-новостные

2. Языки программирования:
 - а) фурнитурные

- б) структурные
- в) фактурные

3. Языки программирования:

- а) управленческие
- б) основные
- в) процедурные

4. Языки программирования:

- а) логические
- б) главные
- в) приобретенные

5. Языки программирования:

- а) объектно-созидательные
- б) объектно-направленные
- в) объектно-ориентированные

6. Языки программирования:

- а) деструктивные
- б) функциональные
- в) конструктивные

7. Языки программирования:

- а) максипарадигмальные
- б) минипарадигмальные
- в) мультипарадигмальные

8. Формальные языки, предназначенный для записи компьютерных программ:

- а) языки программирования
- б) языки передачи
- в) языки записи

9. Программная единица, имеющая имя, по которому она может быть вызвана из других частей программы:

- а) файл
- б) программа
- в) подпрограмма

10. У формальных и фактических параметров должны совпадать:

- а) вид, очередность
- б) количество, порядок
- в) качество, последовательность

11. У формальных и фактических параметров должны совпадать:

- а) следования, тип
- б) последовательность, вид
- в) очередность, форма

12. Для многих широко распространённых языков программирования созданы:

- а) региональные стандарты
- б) международные стандарты
- в) внутренние стандарты

13. Один из наиболее известных языков программирования, используется для обучения программированию в старших классах и на первых курсах вузов, является основой для ряда других языков:

- а) паскаль
- б) пролог
- в) меркьюри

14. Переменные, описанные в основной программе, являются по отношению к внутренним процедурам и функциям:

- а) внутренними
- б) формальными
- в) глобальными

15. Так называется последовательность символов char, сгруппированных в строки, заканчивающиеся специальным символом eoln:

- а) текстовый файл
- б) физический файл
- в) типизированный файл

16. Эта процедура создаёт и открывает новый файл для последующей записи данных:

- а) erase
- б) rewrite
- в) reset

17. Укажите правильно организованное описание переменных в Паскале:

- а) VARIANTS a,b,c:real; i,k:integer;
- б) AR a,b,c:= real; i,k:=integer;
- в) VAR a,b,c:real; i,k:integer;

18. Раздел описания переменных в Паскаль начинается со служебного слова:

- а) LABEL
- б) VAR
- в) CONST

19. Операторы ввода данных:

- а) repeat
- б) Write
- в) Read

20. Укажите целочисленный тип переменных в программе Паскаль:

- а) DOUBLE
- б) INTEGER
- в) REAL

21. Как называется файл, к элементам которого доступ выполняется в той же последовательности, в какой они записывались:

- а) последовательного доступа
- б) элементарного доступа
- в) прямого доступа

22. Файловые переменные Input и Output являются:

- а) логическими
- б) текстовыми
- в) физическими

23. Один из структурных языков программирования:

- а) Scilab
- б) Euphoria
- в) Pascal

24. Один из структурных языков программирования:

- а) Maple
- б) Basic
- в) Limbo

25. Один из структурных языков программирования:

- а) Euphoria
- б) REXX
- в) Алгол

26. Один из процедурных языков программирования:

- а) Оберон
- б) Алгол
- в) Prolog

27. Один из процедурных языков программирования:

- а) Euphoria
- б) Mercury
- в) Dylan

28. Один из процедурных языков программирования:

- а) Оберон
- б) Delphi
- в) Lua

29. Один из объективно-ориентировочных языков программирования:

- а) Dylan
- б) AspectJ
- в) JOVIAL

30. Один из объективно-ориентировочных языков программирования:

- а) QBASIC

б) Delphi

в) Фокал

Критерии и шкала оценивания выполнения тестовых заданий

Таблица 17

Показатели и шкала оценивания тестовых заданий

Текущая аттестация	Количество баллов	Шкала оценивания
выполнение требований по текущей аттестации в полном объеме	90% - 100%	5
	80% - 89%	4
выполнение требований по текущей аттестации в неполном объеме	60% - 79%	3
невыполнение требований по текущей аттестации	менее 60%	2

Перевод набранных при тестировании баллов в оценку производится в соответствии с Положением о фондах оценочных средств для проведения текущего контроля, промежуточной аттестации и государственной итоговой аттестации обучающихся по программам высшего образования.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Вопросы для подготовки к экзамену

1. Языки программирования первого поколения.
2. Языки программирования второго поколения.
3. Языки программирования третьего поколения.
4. Языки программирования четвертого поколения.
5. Языки программирования пятого поколения.
6. Классификация ЯП
7. Алгоритмическое программирование
8. Структурное программирование
9. Основные понятия объектно-ориентированного программирования (ООП). Понятие класса и объекта. Динамические указатели на объекты.
10. Основные понятия ООП: инкапсуляция, наследование, полиморфизм.
11. Основные понятия ООП: поля, методы, свойства.
12. Основные понятия ООП. Четыре секции в объявлении класса, разграничение прав доступа к полям и свойствам класса.
13. ООП. Конструктор и деструктор. Размещение объектов в динамической памяти.
14. ООП. Наследование. Стандартные классы.
15. ООП. Полиморфизм: статический и динамический.
16. Обработка исключительных ситуаций.
17. Переопределение операций
18. Интерфейсы
19. Абстрактные классы
20. Виртуальные методы

Таблица 4

Показатели, критерии и шкала оценивания письменных ответов на экзамене

Критерии оценивания	Показатели и шкала оценивания			
	5	4	3	2
текущая аттестация	выполнение требований по текущей аттестации в полном объеме		выполнение требований по текущей аттестации в неполном объеме	невыполнение требований по текущей аттестации
полнота и правильность ответа	обучающийся полно излагает материал, дает правильное определение основных понятий	обучающийся достаточно полно излагает материал, однако допускает 1-2 ошибки, которые сам же исправляет, и 1-2 недочета в последовательности и языковом оформлении излагаемого	обучающийся демонстрирует знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил	обучающийся демонстрирует незнание большей части соответствующего вопроса
степень осознанности, понимания изученного	демонстрирует понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные	присутствуют 1-2 недочета в обосновании своих суждений, количество приводимых примеров ограничено	не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры	допускает ошибки в формулировке определений и правил, искажающие их смысл
языковое оформление ответа	излагает материал последовательно и правильно с точки зрения норм литературного языка	излагает материал последовательно, с 2-3 ошибками в языковом оформлении	излагает материал непоследовательно и допускает много ошибок в языковом оформлении излагаемого	беспорядочно и неуверенно излагает материал

Примеры тем для курсовой работы

Теоретическая часть

1. Языки программирования: история, концепции и стили программирования, преодоление семантического разрыва.
2. Основные понятия объектно-ориентированного программирования (ООП).
3. Понятие модуля, его структура. Назначение и использование.

4. Алгоритм и его свойства.
5. Способы представления алгоритмов.
6. Понятие рекурсии. Примеры. Достоинства и недостатки рекурсивных алгоритмов.
7. Тестирование и отладка программ.
8. Описание простейшего конечного автомата.
9. Основные понятия ООП. Четыре секции в объявлении класса, разграничение прав доступа к полям и свойствам класса

Практическая часть

1. Описать класс Student, содержащий следующие поля:

- фамилия и инициалы,
- номер группы,
- успеваемость (массив из пяти оценок).

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по номеру группы,
- выводить всех студентов, чей средний балл выше 4,0,
- осуществлять поиск по фамилии студента,
- удалять любого студента,
- редактировать информацию о любом студенте,
- загружать информацию о студентах из файла и сохранять ее в файл.

1. Описать класс Student, содержащий следующие поля:

- фамилия и инициалы,
- номер группы,
- успеваемость (массив из пяти оценок).

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив среднему баллу,
- выводить всех студентов, которые учатся только на 4 и 5,
- осуществлять поиск по номеру группы студента,
- удалять любого студента,
- редактировать информацию о любом студенте,
- загружать информацию о студентах из файла и сохранять ее в файл.

2. Описать класс Student, содержащий следующие поля:

- фамилия и инициалы,
- номер группы,
- успеваемость (массив из пяти оценок).

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по алфавиту,
- выводить всех студентов, имеющих как минимум одну двойку,
- осуществлять поиск по фамилии и инициалам студента,
- удалять любого студента,
- редактировать информацию о любом студенте,
- загружать информацию о студентах из файла и сохранять ее в файл.

3. Описать класс Aeroflot, содержащий следующие поля:

- название пункта назначения,
- название пункта отправления,
- номер рейса,
- тип самолета.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по номеру рейса,
- выводить все рейсы, летящие в выбранный пункт назначения,
- осуществлять поиск по номеру рейса,
- удалять любой рейс,
- редактировать информацию о любом рейсе,
- загружать информацию о рейсах из файла и сохранять ее в файл.

4. Описать класс `Aeroflot`, содержащий следующие поля:

- название пункта назначения,
- название пункта отправления,
- номер рейса,
- тип самолета.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по типу самолета,
- выводить все рейсы, летящие из выбранного пункта отправления,
- осуществлять поиск по пункту назначения,
- удалять любой рейс,
- редактировать информацию о любом рейсе,
- загружать информацию о рейсах из файла и сохранять ее в файл.

5. Описать класс `Aeroflot`, содержащий следующие поля:

- название пункта назначения,
- название пункта отправления,
- номер рейса,
- тип самолета.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по пункту отправления,
- выводить все рейсы, имеющие выбранный тип самолета,
- осуществлять поиск по пункту назначения,
- удалять любой рейс,
- редактировать информацию о любом рейсе,
- загружать информацию о рейсах из файла и сохранять ее в файл.

6. Описать класс `Worker`, содержащий следующие поля:

- фамилия и инициалы,
- должность,
- стаж работы.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по алфавиту,
- выводить всех сотрудников, имеющих указанную должность,
- осуществлять поиск по стажу работы,
- удалять любого сотрудника,
- редактировать информацию о любом сотруднике,
- загружать информацию о сотрудниках из файла и сохранять ее в файл.

7. Описать класс `Worker`, содержащий следующие поля:

- фамилия и инициалы,
- должность,
- стаж работы.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по должности,
- выводить всех сотрудников, имеющих стаж не менее указанного,
- осуществлять поиск по фамилии сотрудника,
- удалять любого сотрудника,
- редактировать информацию о любом сотруднике,
- загружать информацию о сотрудниках из файла и сохранять ее в файл.

8. Описать класс Worker, содержащий следующие поля:

- фамилия и инициалы,
- должность,
- стаж работы.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив стажу по возрасти,
- выводить всех сотрудников, имеющих стаж в диапазоне от А до В,
- осуществлять поиск по фамилии сотрудника,
- удалять любого сотрудника,
- редактировать информацию о любом сотруднике,
- загружать информацию о сотрудниках из файла и сохранять ее в файл.

9. Описать класс Train, содержащий следующие поля:

- номер поезда,
- количество вагонов,
- пункт назначения.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по пункту назначения по алфавиту,
- выводить все поезда, в состав которых входит не менее А вагонов,
- осуществлять поиск по номеру поезда,
- удалять любой поезд,
- редактировать информацию о любом поезде,
- загружать информацию о поездах из файла и сохранять ее в файл.

10. Описать класс Train, содержащий следующие поля:

- номер поезда,
- количество вагонов,
- пункт назначения.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по количеству вагонов,
- выводить все поезда, идущие в выбранный пункт назначения,
- осуществлять поиск по номеру поезда,
- удалять любой поезд,
- редактировать информацию о любом поезде,
- загружать информацию о поездах из файла и сохранять ее в файл.

11. Описать класс Train, содержащий следующие поля:

- номер поезда,
- количество вагонов,
- пункт назначения.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по номеру поезда,

- выводить все поезда, в состав которых входит не более А вагонов,
- осуществлять поиск по пункту назначения,
- удалять любой поезд,
- редактировать информацию о любом поезде,
- загружать информацию о поездах из файла и сохранять ее в файл.

12. Описать класс Marsh, содержащий следующие поля:

- начало маршрута,
- конец маршрута,
- продолжительность маршрута.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по началу маршрута,
- выводить маршруты, имеющие указанную точку завершения маршрута,
- осуществлять поиск по продолжительности маршрута,
- удалять любой маршрут,
- редактировать информацию о любом маршруте,
- загружать информацию о маршрутах из файла и сохранять ее в файл.

13. Описать класс Marsh, содержащий следующие поля:

- начало маршрута,
- конец маршрута,
- продолжительность маршрута.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по точке завершения маршрута,
- выводить маршруты, имеющие указанную точку начала маршрута,
- осуществлять поиск по началу маршрута,
- удалять любой маршрут,
- редактировать информацию о любом маршруте,
- загружать информацию о маршрутах из файла и сохранять ее в файл

14. Описать класс Marsh, содержащий следующие поля:

- начало маршрута,
- конец маршрута,
- продолжительность маршрута.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по точке завершения маршрута,
- выводить маршруты, имеющие указанную длительность маршрута,
- осуществлять поиск по точке завершения маршрута,
- удалять любой маршрут,
- редактировать информацию о любом маршруте,
- загружать информацию о маршрутах из файла и сохранять ее в файл

15. Описать класс Note, содержащий следующие поля:

- номер телефона,
- имя и фамилия,
- дата рождения.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по алфавиту,
- выводить информацию о людях, рожденных в выбранном месяце,
- осуществлять поиск по номеру телефона,

- удалять информацию о любом человеке,
- редактировать информацию о любом человеке,
- загружать информацию о людях из файла и сохранять ее в файл

16. Описать класс Note, содержащий следующие поля:

- номер телефона,
- имя и фамилия,
- дата рождения.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по дате рождения,
- выводить информацию о людях, рожденных в выбранном году,
- осуществлять поиск по фамилии,
- удалять информацию о любом человеке,
- редактировать информацию о любом человеке,
- загружать информацию о людях из файла и сохранять ее в файл

17. Описать класс Note, содержащий следующие поля:

- номер телефона,
- имя и фамилия,
- дата рождения.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по году рождения,
- выводить информацию о людях с выбранной фамилией,
- осуществлять поиск по дате рождения,
- удалять информацию о любом человеке,
- редактировать информацию о любом человеке,
- загружать информацию о людях из файла и сохранять ее в файл

18. Описать класс Znak, содержащий следующие поля:

- знак зодиака,
- имя и фамилия,
- дата рождения.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по фамилии,
- выводить информацию о людях, рожденных в выбранном месяце,
- осуществлять поиск по знаку зодиака,
- удалять информацию о любом человеке,
- редактировать информацию о любом человеке,
- загружать информацию о людях из файла и сохранять ее в файл.

19. Описать класс Znak, содержащий следующие поля:

- знак зодиака,
- имя и фамилия,
- дата рождения.

Написать программу, позволяющую:

- создать массив из 10 экземпляров класса,
- предоставить возможность упорядочить массив по знаку зодиака,
- выводить информацию о людях, рожденных в выбранном году,
- осуществлять поиск по фамилии,
- удалять информацию о любом человеке,
- редактировать информацию о любом человеке,

- загружать информацию о людях из файла и сохранять ее в файл.

Критерии оценки курсовой работы

Критерии оценивания:

- полнота и правильность ответа;
- степень осознанности, понимания изученного;
- языковое оформление ответа.

Таблица 5

Показатели и шкала оценивания выполнения курсовой работы

Шкала оценивания	Показатели
5	работа выполнена без ошибок, обучающийся представил оригинальное и грамотное решение, четко и грамотно оформляет пояснительную записку без отступлений от требований к её оформлению, подробно и безошибочно отвечает на все заданные ему вопросы, проявляет при работе достаточную самостоятельность
4	работа выполнена с незначительными ошибками, но при опросе обучающийся проявляет понимание ошибок и способов их исправления, не допускает существенных погрешностей в ответах на вопросы, аккуратно выполняет демонстрационный материал и пояснительную записку
3	работа выполнена без грубых ошибок, но при опросе обучающийся проявляет недостаточное понимание всех подробностей проделанной работы; допускает при ответах на вопросы неточности и неправильные формулировки; допускает небрежность в графической работе и в оформлении пояснительной записки.
2	принципиальные ошибки в представленной к защите работе и обучающийся при ответах на вопросы, не может устранить указанные недостатки, небрежно выполняет работу и представляет неполную и не соответствующую правилам оформления пояснительную записку, проявляет полное пренебрежение к срокам выполнения проекта.

При обучении с применением дистанционных технологий и электронного обучения промежуточная аттестация проводится в форме компьютерного тестирования в СДО. Оценивание компетентности обучающегося по установленным для дисциплины индикаторам может осуществляться с помощью банка заданий, включающих тестовые задания пяти типов:

- 1 – тестовое задание открытого типа;предусматривающее развернутый ответ обучающегося в нескольких предложениях, составленное с использованием вопросов для подготовки к зачету или экзамену;
- 2 – выбор одного правильного варианта из предложенных вариантов ответов;
- 3 – выбор 2-3 правильных вариантов из предложенных вариантов ответов;

- 4 – установление правильной последовательности в предложенных вариантах ответов/расчётные задачи, ответом на которые будет являться некоторое числовое значение;
- 5 – установление соответствия между двумя множествами вариантов ответов.

Компетенция: ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

Индикатор: ОПК-6.1 Разработка алгоритмов, пригодных для практического использования

Тип задания	Примеры тестовых заданий
1	Какие факторы важны при разработке алгоритмов для практического использования?
1	Какое значение имеет сложность алгоритма в практическом программировании?
1	Почему важно учитывать ограничения ресурсов при разработке алгоритмов?
1	Как роль тестирования алгоритма связана с его практическим использованием?
2	Выберите основной способ, с помощью которого реализуется принцип инкапсуляции в C# Использование классов-наследников и классов-родителей Использование модификаторов доступа, например, модификатора private Использование разных реализаций одного и того же метода у наследника и родителя Ничего из перечисленного
3	Что определяет каждый ЯП? Набор синтаксических правил построение программ как последовательности действий над данными Набор семантических правил Набор лексических правил
4	Установите верную последовательность описания метода класса public Имя_метода int (список параметров)
5	Установите соответствие между термином и его определением А – класс Б – Объект В – Поле Г – Метод 1 - тип данных, определяемый пользователем

	2 - экземпляр класса 3 - элемент данных класса 4 - процедуры и функции, связанные с классом
--	---

Индикатор: ОПК-6.2 Разработка компьютерных программ, пригодных для практического использования

Тип задания	Примеры тестовых заданий
1	Как влияет выбор языка программирования на разработку практических программ?
1	Почему важно учитывать поддержку сообщества при выборе языка программирования?
1	Какие аспекты языка программирования следует учитывать для обеспечения безопасности программ?
1	Как роль документации и читаемости кода влияет на долгосрочное использование программы?
2	Как правильно осуществить наследование класса Manager от класса Worker? class Manager : Worker class Worker: Manager class Manager: class Worker Ничего из перечисленного
3	Что является обязательным условием для реализации конструктора Он должен быть помечен модификатором private Он должен быть помечен модификатором public Имя конструктора должно совпадать с именем класса Имя конструктора должно совпадать с именем наследника класса
4	Установите верную последовательность задания метода класса спецификаторы Имя метода атрибуты тип
5	Установите соответствие между синтаксической конструкцией и ее описанием А – Закрытое поле Б – Открытый метод В – Индексатор Г – Конструктор 1 – private int Pj; 2 – public int Pj(); 3 – public int this [int i] 4 – public Pj();

Составитель: к.ф.-м.н., доцент Черняева С. Н.

Зав. кафедрой: к.ф.-м.н., доцент Черняева С. Н.