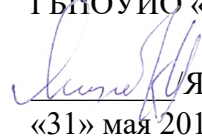




Министерство образования Иркутской области
Государственное бюджетное профессиональное
образовательное учреждение Иркутской области
«Иркутский авиационный техникум»

УТВЕРЖДАЮ
И.О. директора
ГБПОУИО «ИАТ»

 Якубовский А.Н.
«31» мая 2017 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

ОП.12 Объектно-ориентированное программирование

специальности

09.02.03 Программирование в компьютерных системах

Иркутск, 2017

Рассмотрена
цикловой комиссией
ПКС протокол № 12 от
19.05.2017 г.

Председатель ЦК

 /М.А. Кудрявцева /

№	Разработчик ФИО
1	Удальцов Сергей Александрович

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Область применения фонда оценочных средств (ФОС)

ФОС по дисциплине является частью программы подготовки специалистов среднего звена по специальности 09.02.03 Программирование в компьютерных системах

1.2. Место дисциплины в структуре ППССЗ:

ОП.00 Общепрофессиональный цикл.

1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины

В результате освоения дисциплины обучающийся должен	№ дидактической единицы	Формируемая дидактическая единица
Знать	1.1	общие принципы построения алгоритмов, основные алгоритмические конструкции
	1.2	понятие системы программирования
	1.3	основные элементы процедурного языка программирования, структуру программы, операторы и операции;
	1.4	управляющие структуры, структуры данных;
	1.5	объектно-ориентированную модель программирования;
	1.6	понятие классов и объектов, их свойств и методов
	1.7	понятие инкапсуляции
	1.8	понятие полиморфизма
	1.9	понятие механизма наследования классов
Уметь	2.1	использовать языки программирования
	2.2	строить логически правильные эффективные программы
	2.3	создавать классы и объекты на их базе

1.4. Формируемые компетенции:

- 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 в команде, эффективно общаться с коллегами, руководством, потребителями.
- ОК.7 Брать на себя ответственность за работу членов команды (подчиненных), за 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. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ, ИСПОЛЬЗУЕМЫЙ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

2.1 Текущий контроль (ТК) № 1

Тема занятия: 2.5.3.Лабораторная работа №3. Классы и объекты в Java.

Метод и форма контроля: Письменный опрос (Опрос)

Вид контроля: Практическая работа

Дидактическая единица: 1.5 объектно-ориентированную модель программирования;

Занятие(-я):

1.2.1.Объект. Данные объектов. Поведения объектов. Создание объектов.

1.2.6.Объектно-ориентированный подход

2.5.2.Наследование. Суперклассы и подклассы. Переопределение методов. Правила видимости. Зарезервированное слово super.

Задание №1

Ответить на вопросы:

Дать определение, охарактеризовать объектно-ориентированное программирование?

Перечислить предпосылки появления и развития объектно ориентированного программирования.

Перечислить базовые принципы ООП?

Перечислить преимущества ООП.

Описать структуру программы на языке Java. Письменно привести пример простейшей программы.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на часть вопросов. Допущены ошибки.

Задание №2

Ответить на вопросы:

Дать определение, охарактеризовать объектно-ориентированное программирование?

Перечислить предпосылки появления и развития объектно ориентированного программирования.

Перечислить базовые принципы ООП?

Перечислить преимущества ООП.

Описать структуру программы на языке Java. Письменно привести пример

простейшей программы.

<i>Оценка</i>	<i>Показатели оценки</i>
----------------------	---------------------------------

Дидактическая единица: 1.6 понятие классов и объектов, их свойств и методов
Занятие(-я):

1.2.1.Объект. Данные объектов. Поведения объектов. Создание объектов.

1.2.2.Класс. UML диаграммы классов. Атрибуты. Методы. Сообщения.

1.2.6.Объектно-ориентированный подход

2.1.3.Инкапсуляция. Работа со ссылочными переменными. Проекты. Пакеты.
Уровни видимости классов. Базовые пакеты и классы Java.

2.5.2.Наследование. Суперклассы и подклассы. Переопределение методов. Правила видимости. Зарезервированное слово super.

Задание №1

Ответить на вопросы:

Дать определение объекту в ООП? Состояние, поведение.

Дать определение понятиям идентичность и жизненный цикл объектов.

Охарактеризовать взаимоотношения между объектами.

Что такое класс в ООП?

Что такое структура класса? Охарактеризовать абстрактные классы и интерфейсы.

Охарактеризовать отношения между классами. Ассоциация и агрегация.

Охарактеризовать иерархии классов.

В чем разница между классом и объектом?

Что такое атрибуты класса?

Методы класса дать описание. Что такое вызов метода?

Что такое UML диаграммы классов?

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на часть вопросов. Допущены ошибки.

Задание №2

Ответить на вопросы:

Дать определение объекту в ООП? Состояние, поведение.

Дать определение понятиям идентичность и жизненный цикл объектов.
 Охарактеризовать взаимоотношения между объектами.
 Что такое класс в ООП?
 Что такое структура класса? Охарактеризовать абстрактные классы и интерфейсы.
 Охарактеризовать отношения между классами. Ассоциация и агрегация.
 Охарактеризовать иерархии классов.
 В чем разница между классом и объектом?
 Что такое атрибуты класса?
 Методы класса дать описание. Что такое вызов метода?
 Что такое UML диаграммы классов?

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

2.2 Текущий контроль (ТК) № 2

Тема занятия: 2.7.7.Лабораторная работа №7. Работа с файлами.

Метод и форма контроля: Письменный опрос (Опрос)

Вид контроля: Практическая работа

Дидактическая единица: 1.1 общие принципы построения алгоритмов, основные алгоритмические конструкции

Занятие(-я):

1.2.6.Объектно-ориентированный подход

2.1.2.Зарезервированные слова и литералы Java. Управляющие последовательности. Идентификаторы. Переменные и типы. Примитивные и ссылочные типы.

2.2.1.Создание в NetBeans простейшего приложения Java. Компиляция файлов проекта, запуск приложения. Структура проекта NetBeans.

2.2.2.Лабораторная работа №1. Разработка программ на Java.

2.4.2.Особенности целочисленных вычислений - организация циклов, приоритет операторов и арифметическое переполнение. Оператор цикла while/ do while. Операторы прерывания continue, break, return, system.exit

2.4.3.Лабораторная работа №2. Типы данных и управляющие структуры Java.

2.5.1.Функции. Модификаторы. Передача примитивных типов в функции. Локальные глобальные переменные. Модификаторы доступа. Правила видимости. Ссылка this. Передача ссылочных типов в функции.

2.5.3.Лабораторная работа №3. Классы и объекты в Java.

2.5.4.Статическое и динамическое связывание методов. Полиморфизм. Базовый класс Object. Конструкторы. Зарезервированные слова super и this. Блоки инициализации. Удаление неиспользуемых объектов, метод finalize. Проблема деструкторов. Перегрузка методов. Правила совместимости ссылочных типов как основа использования полиморфного кода.

2.6.1.UML - диаграммы. Панель инструментов проектов с UML - диаграммами.

Прямое проектирование - построение кода классов по UML - диаграммам.
Обратное проектирование - построение UML диаграмм по разработанным классам.
Рефакторинг.

2.7.1.Массивы. Коллекции, списки, итераторы. Перебор в цикле элементов коллекций. Работа со строками, строки, как объекты. Классы String, StringBuffer и StringBuilder. Типы перечисления. Работа с датами и временем.

2.7.2.Лабораторная работа №5. Коллекции.

2.7.4.Работа с графикой и графическим пользовательским интерфейсом.
Графические примитивы.

2.7.5.Исключительные ситуации. Обработка исключительных ситуаций. Иерархия исключительных ситуаций. Объявление типа исключительной ситуации и оператор throw. Метод, возбуждающий исключительную ситуацию.

2.7.6.Работа с файлами и папками. Работа с потоками ввода/вывода.

Задание №1

Блок задания 1

Дать описание следующим принципам:

Принцип поэтапной детализации алгоритма.

Принцип "от главного к второстепенному"

Принцип структурирования.

Дать определение: понятие, суждение, умозаключение, высказывание, предикат.

Основные алгоритмические конструкции: следование, ветвление, цикл. Нарисовать схему, привести пример.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на некоторые вопросы. Присутствует неполнота ответов. Ответы не отражают целиком суть вопроса.

Задание №2

Блок задания 1

Дать описание следующим принципам:

Принцип поэтапной детализации алгоритма.

Принцип "от главного к второстепенному"

Принцип структурирования.

Дать определение: понятие, суждение, умозаключение, высказывание, предикат. Основные алгоритмические конструкции: следование, ветвление, цикл. Нарисовать схему, привести пример.

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

Дидактическая единица: 1.2 понятие системы программирования

Занятие(-я):

1.1.1.Фундаментальные концепции. Переход с процедурной разработки на объектно-ориентированную. Процедурное программирование. Объектно-ориентированное программирование.

1.2.6.Объектно-ориентированный подход

2.1.1.Java и другие языки программирования. Категории программ, написанных на Java. Алфавит языка Java.

2.1.3.Инкапсуляция. Работа со ссылочными переменными. Проекты. Пакеты. Уровни видимости классов. Базовые пакеты и классы Java.

2.1.4.Технологии Java, .NET, ASP, PHP. Среды разработки NetBeans, Eclipse, JDeveloper, JBuilder, IntelliJ IDEA.

2.2.2.Лабораторная работа №1. Разработка программ на Java.

2.3.1.Логический тип. Целые типы переменные, константы. Основные операторы. Вещественные типы и класс Math. Упаковка (boxing) и распаковка (unboxing). Приоритет операторов.

2.4.1.Составной оператор. Условный оператор if. Оператор выбора switch. Операторы инкремента ++ и декремента -- Оператор цикла for. Ошибки при использовании вещественного счётчика цикла. Эффективная организация циклов при вычислениях в формате с плавающей точкой.

2.4.3.Лабораторная работа №2. Типы данных и управляющие структуры Java.

2.5.1.Функции. Модификаторы. Передача примитивных типов в функции. Локальные глобальные переменные. Модификаторы доступа. Правила видимости. Ссылка this. Передача ссылочных типов в функции.

2.5.2.Наследование. Суперклассы и подклассы. Переопределение методов. Правила видимости. Зарезервированное слово super.

2.5.3.Лабораторная работа №3. Классы и объекты в Java.

2.5.4.Статическое и динамическое связывание методов. Полиморфизм. Базовый класс Object. Конструкторы. Зарезервированные слова super и this. Блоки инициализации. Удаление неиспользуемых объектов, метод finalize. Проблема деструкторов. Перегрузка методов. Правила совместимости ссылочных типов как основа использования полиморфного кода.

2.7.1.Массивы. Коллекции, списки, итераторы. Перебор в цикле элементов коллекций. Работа со строками, строки, как объекты. Классы String, StringBuffer и StringBuilder. Типы перечисления. Работа с датами и временем.

2.7.2.Лабораторная работа №5. Коллекции.

2.7.4.Работа с графикой и графическим пользовательским интерфейсом.

Графические примитивы.

2.7.5.Исключительные ситуации. Обработка исключительных ситуаций. Иерархия исключительных ситуаций. Объявление типа исключительной ситуации и оператор throw. Метод, возбуждающий исключительную ситуацию.

2.7.6.Работа с файлами и папками. Работа с потоками ввода/вывода.

Задание №1

Блок задания 2

Перечислить категории программ на Java, дать описание.

Зарезервированные слова. Привести примеры. Литералы.

Что такое управляющие последовательности?

Дать определение понятию идентификатора. Краткое описание. Примитивный и ссылочные типы.

Привести примеры базовых пакетов и классов Java с кратким описанием.

Примитивные типы данных. Описание, характеристики.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на некоторые вопросы. Присутствует неполнота ответов. Ответы не отражают целиком суть вопроса.

Задание №2

Блок задания 2

Перечислить категории программ на Java, дать описание.

Зарезервированные слова. Привести примеры. Литералы.

Что такое управляющие последовательности?

Дать определение понятию идентификатора. Краткое описание. Примитивный и ссылочные типы.

Привести примеры базовых пакетов и классов Java с кратким описанием.

Примитивные типы данных. Описание, характеристики.

<i>Оценка</i>	<i>Показатели оценки</i>
----------------------	---------------------------------

Дидактическая единица: 1.3 основные элементы процедурного языка программирования, структуру программы, операторы и операции;

Занятие(-я):

1.2.6.Объектно-ориентированный подход

2.4.1.Составной оператор. Условный оператор if. Оператор выбора switch.

Операторы инкремента ++ и декремента -- Оператор цикла for. Ошибки при использовании вещественного счётчика цикла. Эффективная организация циклов при вычислениях в формате с плавающей точкой.

2.4.3.Лабораторная работа №2. Типы данных и управляющие структуры Java.

2.7.1.Массивы. Коллекции, списки, итераторы. Перебор в цикле элементов коллекций. Работа со строками, строки, как объекты. Классы String, StringBuffer и StringBuilder. Типы перечисления. Работа с датами и временем.

2.7.4.Работа с графикой и графическим пользовательским интерфейсом.

Графические примитивы.

2.7.5.Исключительные ситуации. Обработка исключительных ситуаций. Иерархия исключительных ситуаций. Объявление типа исключительной ситуации и оператор throw. Метод, возбуждающий исключительную ситуацию.

2.7.6.Работа с файлами и папками. Работа с потоками ввода/вывода.

Задание №1

Блок задания 3

Дать определение понятию процедурного языка программирования.

Структура программы на процедурном языке.

Сравнить процедурную парадигму и объектно-ориентированную.

Перечислить основные операторы и операции процедурного языка программирования.

Преимущества процедурного подхода.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на некоторые вопросы. Присутствует неполнота ответов. Ответы не отражают целиком суть вопроса.

Задание №2

Блок задания 3

Дать определение понятию процедурного языка программирования.
 Структура программы на процедурном языке.
 Сравнить процедурную парадигму и объектно-ориентированную.
 Перечислить основные операторы и операции процедурного языка программирования.
 Преимущества процедурного подхода.

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

Дидактическая единица: 1.4 управляющие структуры, структуры данных;

Занятие(-я):

1.2.6.Объектно-ориентированный подход

2.4.2.Особенности целочисленных вычислений - организация циклов, приоритет операторов и арифметическое переполнение. Оператор цикла while/ do while.

Операторы прерывания continue, break, return, system.exit

2.4.3.Лабораторная работа №2. Типы данных и управляющие структуры Java.

2.5.3.Лабораторная работа №3. Классы и объекты в Java.

2.6.1.UML - диаграммы. Панель инструментов проектов с UML - диаграммами.

Прямое проектирование - построение кода классов по UML - диаграммам.

Обратное проектирование - построение UML диаграмм по разработанным классам.

Рефакторинг.

2.7.4.Работа с графикой и графическим пользовательским интерфейсом.

Графические примитивы.

2.7.5.Исключительные ситуации. Обработка исключительных ситуаций. Иерархия исключительных ситуаций. Объявление типа исключительной ситуации и оператор throw. Метод, возбуждающий исключительную ситуацию.

2.7.6.Работа с файлами и папками. Работа с потоками ввода/вывода.

Задание №1

Блок задания 4

Что такое управляющие конструкции?

Составной, условный оператор. Описание, пример кода.

Оператор выбора. Описание, пример кода.

Операторы цикла. Перечислить, привести примеры использования в коде.

Операторы прерывания. Описание функции. Примеры использования.

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

Задание №2

Блок задания 4

Что такое управляющие конструкции?

Составной, условный оператор. Описание, пример кода.

Оператор выбора. Описание, пример кода.

Операторы цикла. Перечислить, привести примеры использования в коде.

Операторы прерывания. Описание функции. Примеры использования.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на некоторые вопросы. Присутствует неполнота ответов. Ответы не отражают целиком суть вопроса.

Дидактическая единица: 2.3 создавать классы и объекты на их базе

Занятие(-я):

2.5.3.Лабораторная работа №3. Классы и объекты в Java.

2.5.4.Статическое и динамическое связывание методов. Полиморфизм. Базовый класс Object. Конструкторы. Зарезервированные слова super и this. Блоки инициализации. Удаление неиспользуемых объектов, метод finalize. Проблема деструкторов. Перегрузка методов. Правила совместимости ссылочных типов как основа использования полиморфного кода.

Задание №1

Составить диаграмму классов в соответствии с индивидуальным заданием.

<i>Оценка</i>	<i>Показатели оценки</i>
----------------------	---------------------------------

Задание №2

Составить диаграмму классов в соответствии с индивидуальным заданием.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Диаграмма составлена в соответствии с индивидуальным заданием.
4	Допущены незначительные ошибки.
3	Допущены существенные ошибки.

2.3 Текущий контроль (ТК) № 3

Тема занятия: 2.10.2.Лабораторная работа №8. Использование коллекций.

Метод и форма контроля: Лабораторная работа (Опрос)

Вид контроля: Самостоятельная работа

Дидактическая единица: 1.7 понятие инкапсуляции

Занятие(-я):

1.2.3.Инкапсуляция и скрытие данных. Интерфейсы и реализации. Модель парадигмы "интерфейс/реализация".

1.2.6.Объектно-ориентированный подход

2.1.3.Инкапсуляция. Работа со ссылочными переменными. Проекты. Пакеты. Уровни видимости классов. Базовые пакеты и классы Java.

Задание №1

Дать развернутое определение понятию инкапсуляция.

Написать пример кода, демонстрирующий применение принципа инкапсуляции.

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

Задание №2

Дать развернутое определение понятию инкапсуляция.

Написать пример кода, демонстрирующий применение принципа инкапсуляции.

<i>Оценка</i>	<i>Показатели оценки</i>
3	Дано только определение.
4	Верно дано определение, приведен пример в виде кода. Допущены ошибки.
5	Верно дано определение. Приведенный пример хорошо отражает суть принципа. Допускаются несущественные неточности.

Дидактическая единица: 1.8 понятие полиморфизма

Занятие(-я):

1.2.5.Полиморфизм. Композиция. Абстрагирование.

1.2.6.Объектно-ориентированный подход

2.5.4.Статическое и динамическое связывание методов. Полиморфизм. Базовый класс Object. Конструкторы. Зарезервированные слова super и this. Блоки инициализации. Удаление неиспользуемых объектов, метод finalize. Проблема деструкторов. Перегрузка методов. Правила совместимости ссылочных типов как основа использования полиморфного кода.

2.5.5.Лабораторная работа №4 Наследование и полиморфизм.

Задание №1

Дать развернутое определение понятию полиморфизма.

Написать пример кода, демонстрирующий принцип полиморфизма.

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

Задание №2

Дать развернутое определение понятию полиморфизма.

Написать пример кода, демонстрирующий принцип полиморфизма.

<i>Оценка</i>	<i>Показатели оценки</i>
3	Дано только определение.
4	Верно дано определение, приведен пример в виде кода. Допущены ошибки.
5	Верно дано определение. Приведенный пример хорошо отражает суть принципа. Допускаются несущественные неточности.

Дидактическая единица: 1.9 понятие механизма наследования классов

Занятие(-я):

1.2.4.Наследование. Суперклассы и подклассы.

1.2.6.Объектно-ориентированный подход

2.5.2.Наследование. Суперклассы и подклассы. Переопределение методов. Правила видимости. Зарезервированное слово super.

2.5.5.Лабораторная работа №4 Наследование и полиморфизм.

2.6.1.UML - диаграммы. Панель инструментов проектов с UML - диаграммами.

Прямое проектирование - построение кода классов по UML - диаграммам.

Обратное проектирование - построение UML диаграмм по разработанным классам.

Рефакторинг.

2.8.1.Проблемы множественного наследования классов. Интерфейсы. Отличия интерфейсов от классов. Проблемы наследования интерфейсов. Композиция - альтернатива множественному наследованию.

Задание №1

Дать развернутое определение понятию наследование.

Написать пример кода, демонстрирующий применение принципа наследования.

<i>Оценка</i>	<i>Показатели оценки</i>
3	Дано только определение.
4	Верно дано определение, приведен пример в виде кода. Допущены ошибки.
5	Верно дано определение. Приведенный пример хорошо отражает суть принципа. Допускаются несущественные неточности.

Дидактическая единица: 2.1 использовать языки программирования

Занятие(-я):

1.2.6.Объектно-ориентированный подход

2.1.1.Java и другие языки программирования. Категории программ, написанных на Java. Алфавит языка Java.

2.1.2.Зарезервированные слова и литералы Java. Управляющие последовательности. Идентификаторы. Переменные и типы. Примитивные и ссылочные типы.

2.1.4.Технологии Java, .NET, ASP, PHP. Среда разработки NetBeans, Eclipse, JDeveloper, JBuilder, IntelliJ IDEA.

2.2.1.Создание в NetBeans простейшего приложения Java. Компиляция файлов проекта, запуск приложения. Структура проекта NetBeans.

2.2.2.Лабораторная работа №1. Разработка программ на Java.

2.4.1.Составной оператор. Условный оператор if. Оператор выбора switch.

Операторы инкремента ++ и декремента -- Оператор цикла for. Ошибки при использовании вещественного счётчика цикла. Эффективная организация циклов при вычислениях в формате с плавающей точкой.

2.4.2.Особенности целочисленных вычислений - организация циклов, приоритет операторов и арифметическое переполнение. Оператор цикла while/ do while.

Операторы прерывания continue, break, return, system.exit

2.4.3.Лабораторная работа №2. Типы данных и управляющие структуры Java.

2.5.3.Лабораторная работа №3. Классы и объекты в Java.

2.5.4.Статическое и динамическое связывание методов. Полиморфизм. Базовый класс Object. Конструкторы. Зарезервированные слова super и this. Блоки инициализации. Удаление неиспользуемых объектов, метод finalize. Проблема деструкторов. Перегрузка методов. Правила совместимости ссылочных типов как основа использования полиморфного кода.

2.5.5.Лабораторная работа №4 Наследование и полиморфизм.

2.7.1.Массивы. Коллекции, списки, итераторы. Перебор в цикле элементов коллекций. Работа со строками, строки, как объекты. Классы String, StringBuffer и StringBuilder. Типы перечисления. Работа с датами и временем.

2.7.2.Лабораторная работа №5. Коллекции.

2.7.4.Работа с графикой и графическим пользовательским интерфейсом.

Графические примитивы.

2.7.6.Работа с файлами и папками. Работа с потоками ввода/вывода.

2.7.7.Лабораторная работа №7. Работа с файлами.

2.9.1.Потоки выполнения (threads) и синхронизация. Преимущества и проблемы при работе с потоками выполнения. Синхронизация по ресурсам и событиям. Класс Thread и интерфейсы Runnable и Callable. Создание и запуск потока выполнения. Поля и методы в классе Thread.

2.10.1.Виды вложенных классов. Статические (static) вложенные классы и

интерфейсы. Внутренние (inner) классы. Локальные (local) классы. Анонимные классы и обработчики событий. Анонимные классы и слушатели событий (listeners)

Задание №1

Самостоятельно написать простую программу, которая демонстрирует три главных принципа ООП. Ответить на вопросы к программе.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Программа написана верно. Даны ответы на все заданные вопросы. Допускаются несущественные ошибки.
4	Программа написана с некоторыми несущественными ошибками. Даны ответы на большее количество вопросов.
3	В программном коде допущены ошибки. Верно даны ответы на некоторые вопросы.

Дидактическая единица: 2.2 строить логически правильные эффективные программы

Занятие(-я):

2.4.1.Составной оператор. Условный оператор if. Оператор выбора switch. Операторы инкремента ++ и декремента -- Оператор цикла for. Ошибки при использовании вещественного счётчика цикла. Эффективная организация циклов при вычислениях в формате с плавающей точкой.

2.4.3.Лабораторная работа №2. Типы данных и управляющие структуры Java.

2.5.4.Статическое и динамическое связывание методов. Полиморфизм. Базовый класс Object. Конструкторы. Зарезервированные слова super и this. Блоки инициализации. Удаление неиспользуемых объектов, метод finalize. Проблема деструкторов. Перегрузка методов. Правила совместимости ссылочных типов как основа использования полиморфного кода.

2.5.5.Лабораторная работа №4 Наследование и полиморфизм.

2.7.2.Лабораторная работа №5. Коллекции.

2.7.7.Лабораторная работа №7. Работа с файлами.

2.9.1.Потоки выполнения (threads) и синхронизация. Преимущества и проблемы при работе с потоками выполнения. Синхронизация по ресурсам и событиям. Класс Thread и интерфейсы Runnable и Callable. Создание и запуск потока выполнения. Поля и методы в классе Thread.

Задание №1

Спроектировать структуру программы.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Спроектированная структура является рабочим решением. Не содержит лишних элементов.

4	Спроектированная структура имеет недостатки. Допущены логические ошибки.
3	Спроектированная структура неверна. Допущены существенные логические ошибки.

3. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ, ИСПОЛЬЗУЕМЫЙ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

№ семестра	Вид промежуточной аттестации
7	Дифференцированный зачет

Дифференцированный зачет может быть выставлен автоматически по результатам текущих контролей
Текущий контроль №1
Текущий контроль №2
Текущий контроль №3

Метод и форма контроля: Практическая работа (Опрос)

Вид контроля: Ответить на два теоретических вопроса и одно практическое задание.

Дидактическая единица для контроля:

1.1 общие принципы построения алгоритмов, основные алгоритмические конструкции

Задание №1 (из текущего контроля)

Блок задания 1

Дать описание следующим принципам:

Принцип поэтапной детализации алгоритма.

Принцип "от главного к второстепенному"

Принцип структурирования.

Дать определение: понятие, суждение, умозаключение, высказывание, предикат.

Основные алгоритмические конструкции: следование, ветвление, цикл. Нарисовать схему, привести пример.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на некоторые вопросы. Присутствует неполнота ответов. Ответы не отражают целиком суть вопроса.

Задание №2 (из текущего контроля)

Блок задания 1

Дать описание следующим принципам:

Принцип поэтапной детализации алгоритма.

Принцип "от главного к второстепенному"

Принцип структурирования.

Дать определение: понятие, суждение, умозаключение, высказывание, предикат.

Основные алгоритмические конструкции: следование, ветвление, цикл. Нарисовать схему, привести пример.

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

Дидактическая единица для контроля:

1.2 понятие системы программирования

Задание №1 (из текущего контроля)

Блок задания 2

Перечислить категории программ на Java, дать описание.

Зарезервированные слова. Привести примеры. Литералы.

Что такое управляющие последовательности?

Дать определение понятию идентификатора. Краткое описание. Примитивный и ссылочные типы.

Привести примеры базовых пакетов и классов Java с кратким описанием.

Примитивные типы данных. Описание, характеристики.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на некоторые вопросы. Присутствует неполнота ответов. Ответы не отражают целиком суть вопроса.

Задание №2 (из текущего контроля)

Блок задания 2

Перечислить категории программ на Java, дать описание.

Зарезервированные слова. Привести примеры. Литералы.

Что такое управляющие последовательности?

Дать определение понятию идентификатора. Краткое описание. Прimitивный и ссылочные типы.

Привести примеры базовых пакетов и классов Java с кратким описанием.

Примитивные типы данных. Описание, характеристики.

<i>Оценка</i>	<i>Показатели оценки</i>
----------------------	---------------------------------

Дидактическая единица для контроля:

1.3 основные элементы процедурного языка программирования, структуру программы, операторы и операции;

Задание №1 (из текущего контроля)

Блок задания 3

Дать определение понятию процедурного языка программирования.

Структура программы на процедурном языке.

Сравнить процедурную парадигму и объектно-ориентированную.

Перечислить основные операторы и операции процедурного языка программирования.

Преимущества процедурного подхода.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на некоторые вопросы. Присутствует неполнота ответов. Ответы не отражают целиком суть вопроса.

Задание №2 (из текущего контроля)

Блок задания 3

Дать определение понятию процедурного языка программирования.

Структура программы на процедурном языке.

Сравнить процедурную парадигму и объектно-ориентированную.

Перечислить основные операторы и операции процедурного языка программирования.

Преимущества процедурного подхода.

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

Дидактическая единица для контроля:

1.4 управляющие структуры, структуры данных;

Задание №1 (из текущего контроля)

Блок задания 4

Что такое управляющие конструкции?

Составной, условный оператор. Описание, пример кода.

Оператор выбора. Описание, пример кода.

Операторы цикла. Перечислить, привести примеры использования в коде.

Операторы прерывания. Описание функции. Примеры использования.

<i>Оценка</i>	<i>Показатели оценки</i>
---------------	--------------------------

Задание №2 (из текущего контроля)

Блок задания 4

Что такое управляющие конструкции?

Составной, условный оператор. Описание, пример кода.

Оператор выбора. Описание, пример кода.

Операторы цикла. Перечислить, привести примеры использования в коде.

Операторы прерывания. Описание функции. Примеры использования.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на некоторые вопросы. Присутствует неполнота ответов. Ответы не отражают целиком суть вопроса.

Дидактическая единица для контроля:

1.5 объектно-ориентированную модель программирования;

Задание №1 (из текущего контроля)

Ответить на вопросы:

Дать определение, охарактеризовать объектно-ориентированное программирование?

Перечислить предпосылки появления и развития объектно ориентированного

программирования.

Перечислить базовые принципы ООП?

Перечислить преимущества ООП.

Описать структуру программы на языке Java. Письменно привести пример простейшей программы.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на часть вопросов. Допущены ошибки.

Задание №2 (из текущего контроля)

Ответить на вопросы:

Дать определение, охарактеризовать объектно-ориентированное программирование?

Перечислить предпосылки появления и развития объектно ориентированного программирования.

Перечислить базовые принципы ООП?

Перечислить преимущества ООП.

Описать структуру программы на языке Java. Письменно привести пример простейшей программы.

<i>Оценка</i>	<i>Показатели оценки</i>
----------------------	---------------------------------

Дидактическая единица для контроля:

1.6 понятие классов и объектов, их свойств и методов

Задание №1 (из текущего контроля)

Ответить на вопросы:

Дать определение объекту в ООП? Состояние, поведение.

Дать определение понятиям идентичность и жизненный цикл объектов.

Охарактеризовать взаимоотношения между объектами.

Что такое класс в ООП?

Что такое структура класса? Охарактеризовать абстрактные классы и интерфейсы.

Охарактеризовать отношения между классами. Ассоциация и агрегация.

Охарактеризовать иерархии классов.

В чем разница между классом и объектом?

Что такое атрибуты класса?

Методы класса дать описание. Что такое вызов метода?

Что такое UML диаграммы классов?

<i>Оценка</i>	<i>Показатели оценки</i>
5	Верно даны ответы на все заданные вопросы. Подробно охарактеризованы понятия и определения. Допускаются незначительные неточности.
4	Верно даны ответы на большую часть вопросов. Правильно передана суть вопросов.
3	Верно даны ответы на часть вопросов. Допущены ошибки.

Задание №2 (из текущего контроля)

Ответить на вопросы:

Дать определение объекту в ООП? Состояние, поведение.

Дать определение понятиям идентичность и жизненный цикл объектов.

Охарактеризовать взаимоотношения между объектами.

Что такое класс в ООП?

Что такое структура класса? Охарактеризовать абстрактные классы и интерфейсы.

Охарактеризовать отношения между классами. Ассоциация и агрегация.

Охарактеризовать иерархии классов.

В чем разница между классом и объектом?

Что такое атрибуты класса?

Методы класса дать описание. Что такое вызов метода?

Что такое UML диаграммы классов?

<i>Оценка</i>	<i>Показатели оценки</i>
----------------------	---------------------------------

Дидактическая единица для контроля:

1.7 понятие инкапсуляции

Задание №1 (из текущего контроля)

Дать развернутое определение понятию инкапсуляция.

Написать пример кода, демонстрирующий применение принципа инкапсуляции.

<i>Оценка</i>	<i>Показатели оценки</i>
----------------------	---------------------------------

Задание №2 (из текущего контроля)

Дать развернутое определение понятию инкапсуляция.

Написать пример кода, демонстрирующий применение принципа инкапсуляции.

Оценка	Показатели оценки
3	Дано только определение.
4	Верно дано определение, приведен пример в виде кода. Допущены ошибки.
5	Верно дано определение. Приведенный пример хорошо отражает суть принципа. Допускаются несущественные неточности.

Дидактическая единица для контроля:

1.8 понятие полиморфизма

Задание №1 (из текущего контроля)

Дать развернутое определение понятию полиморфизма.

Написать пример кода, демонстрирующий принцип полиморфизма.

Оценка	Показатели оценки
---------------	--------------------------

Задание №2 (из текущего контроля)

Дать развернутое определение понятию полиморфизма.

Написать пример кода, демонстрирующий принцип полиморфизма.

Оценка	Показатели оценки
3	Дано только определение.
4	Верно дано определение, приведен пример в виде кода. Допущены ошибки.
5	Верно дано определение. Приведенный пример хорошо отражает суть принципа. Допускаются несущественные неточности.

Дидактическая единица для контроля:

1.9 понятие механизма наследования классов

Задание №1 (из текущего контроля)

Дать развернутое определение понятию наследование.

Написать пример кода, демонстрирующий применение принципа наследования.

Оценка	Показатели оценки
3	Дано только определение.

4	Верно дано определение, приведен пример в виде кода. Допущены ошибки.
5	Верно дано определение. Приведенный пример хорошо отражает суть принципа. Допускаются несущественные неточности.

Дидактическая единица для контроля:

2.1 использовать языки программирования

Задание №1 (из текущего контроля)

Самостоятельно написать простую программу, которая демонстрирует три главных принципа ООП. Ответить на вопросы к программе.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Программа написана верно. Даны ответы на все заданные вопросы. Допускаются несущественные ошибки.
4	Программа написана с некоторыми несущественными ошибками. Даны ответы на большее количество вопросов.
3	В программном коде допущены ошибки. Верно даны ответы на некоторые вопросы.

Дидактическая единица для контроля:

2.2 строить логически правильные эффективные программы

Задание №1 (из текущего контроля)

Спроектировать структуру программы.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Спроектированная структура является рабочим решением. Не содержит лишних элементов.
4	Спроектированная структура имеет недостатки. Допущены логические ошибки.
3	Спроектированная структура неверна. Допущены существенные логические ошибки.

Дидактическая единица для контроля:

2.3 создавать классы и объекты на их базе

Задание №1 (из текущего контроля)

Составить диаграмму классов в соответствии с индивидуальным заданием.

<i>Оценка</i>	<i>Показатели оценки</i>
----------------------	---------------------------------

Задание №2 (из текущего контроля)

Составить диаграмму классов в соответствии с индивидуальным заданием.

<i>Оценка</i>	<i>Показатели оценки</i>
5	Диаграмма составлена в соответствии с индивидуальным заданием.
4	Допущены незначительные ошибки.
3	Допущены существенные ошибки.