

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ПО ДИСЦИПЛИНЕ**
Объектно-ориентированное программирование

Код модуля
1156466(1)

Модуль
Основания информационных технологий II часть

Екатеринбург

Оценочные материалы составлены автором(ами):

№ п/п	Фамилия, имя, отчество	Ученая степень, ученое звание	Должность	Подразделение
1	Домашних Иван Алексеевич	без ученой степени, без ученого звания	Старший преподават ель	департамент математики, механики и компьютерных наук

Согласовано:

Управление образовательных программ

Ю.Д. Маева

Авторы:

- Домашних Иван Алексеевич, Старший преподаватель, департамент математики, механики и компьютерных наук

1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ **Объектно-ориентированное программирование**

1.	Объем дисциплины в зачетных единицах	3	
2.	Виды аудиторных занятий	Лекции Лабораторные занятия	
3.	Промежуточная аттестация	Зачет	
4.	Текущая аттестация	Контрольная работа	2
		Домашняя работа	4

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ **Объектно-ориентированное программирование**

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

Код и наименование компетенции	Планируемые результаты обучения (индикаторы)	Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине
1	2	3
ОПК-4 -Способен решать задачи профессиональной деятельности с использованием информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Д-1 - Демонстрировать развитие компетенций в области ИТ З-2 - Демонстрировать понимание нормативных требований к информационной безопасности У-1 - Обосновывать выбор современных ИТ-технологий для сбора, анализа, обработки и представления информации по профилю деятельности У-2 - Выбирать безопасные информационно-коммуникативные технологии для эффективного решения задач профессиональной деятельности	Домашняя работа № 1 Домашняя работа № 2 Домашняя работа № 3 Домашняя работа № 4 Зачет Контрольная работа № 1 Контрольная работа № 2 Лабораторные занятия Лекции

<p>ОПК-5 -Способен использовать существующие программные продукты и информационные базы данных для решения задач профессиональной деятельности</p>	<p>Д-1 - Демонстрировать развитие компетенций в области ИТ П-2 - Иметь опыт решения задач профессиональной деятельности с использованием современных информационных баз данных У-2 - Осуществлять поиск и выбор необходимых информационных баз данных для решения профессиональных задач</p>	<p>Домашняя работа № 1 Домашняя работа № 2 Домашняя работа № 3 Домашняя работа № 4 Зачет Контрольная работа № 1 Контрольная работа № 2 Лабораторные занятия Лекции</p>
<p>ПК-3 -Способен собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности, а также разрабатывать новые алгоритмические, методические и технологические решения в конкретной сфере профессиональной деятельности</p>	<p>П-1 - Подготовить техническое задание на разработку информационной системы У-1 - Интегрировать в практическую деятельность профессиональные стандарты в области информационных технологий</p>	<p>Домашняя работа № 1 Домашняя работа № 2 Домашняя работа № 3 Домашняя работа № 4 Зачет Контрольная работа № 1 Контрольная работа № 2 Лабораторные занятия Лекции</p>

3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)

3.1. Процедуры текущей и промежуточной аттестации по дисциплине

1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – 0.50		
Текущая аттестация на лекциях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>домашняя работа</i>	4	20
<i>домашняя работа</i>	8	20
<i>домашняя работа</i>	15	20
<i>контрольная работа</i>	10	20

<i>контрольная работа</i>	17	20
Весовой коэффициент значимости результатов текущей аттестации по лекциям – 0.50		
Промежуточная аттестация по лекциям – зачет		
Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – 0.50		
2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – не предусмотрено		
Текущая аттестация на практических/семинарских занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям– не предусмотрено		
Промежуточная аттестация по практическим/семинарским занятиям– нет		
Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям– не предусмотрено		
3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий –0.50		
Текущая аттестация на лабораторных занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>Домашняя работа</i>	16	100
Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям -1.00		
Промежуточная аттестация по лабораторным занятиям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – 0.00		
4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий –не предусмотрено		
Текущая аттестация на онлайн-занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям -не предусмотрено		
Промежуточная аттестация по онлайн-занятиям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – не предусмотрено		

3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

Текущая аттестация выполнения курсовой работы/проекта	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта– не предусмотрено		
Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта– защиты – не предусмотрено		

4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

Критерии оценивания учебных достижений обучающихся

Результаты обучения	Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам
Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.
Другие результаты	Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов. Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения. Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

Шкала оценивания достижения результатов обучения (индикаторов) по уровням

Характеристика уровней достижения результатов обучения (индикаторов)				
№ п/п	Содержание уровня выполнения критерия оценивания результатов обучения (выполненное оценочное задание)	Шкала оценивания		
		Традиционная характеристика уровня		Качественная характеристика уровня
1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)

2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ

5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля

5.1.1. Лекции

Самостоятельное изучение теоретического материала по темам/разделам лекций в соответствии с содержанием дисциплины (п. 1.2. РПД)

5.1.2. Лабораторные занятия

Примерный перечень тем

1. Введение в C#
 2. Инкапсуляция. Теория
 3. Наследование и полиморфизм
 4. Generics. Делегирование
 5. Рефлексия
 6. DDD
 7. FluentAPI
 8. Модульность
 9. Управление зависимостями
 10. DI-контейнеры
 11. Функциональный стиль
 12. Управление ресурсами. Работа с файлами
- LMS-платформа
1. <https://ulearn.me/Course/cs2>

5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля

Разноуровневое (дифференцированное) обучение.

Базовый

5.2.1. Контрольная работа № 1

Примерный перечень тем

1. Инкапсуляция

Примерные задания

Создайте класс Superhero с одним свойством Name. Сделайте так, чтобы у этого класса могло быть не более двух экземпляров: Супермен и Бэтмен. При этом создавались эти экземпляры только по мере первой необходимости. В частности, если понадобился только бэтмен, то экземпляр супермена не должен создаваться.

LMS-платформа – не предусмотрена

5.2.2. Контрольная работа № 2

Примерный перечень тем

1. Generics

Примерные задания

Вы разработали следующий класс строителя лабиринта, использующий идею method chaining:

```
public class MazeBuilder
{
    protected Maze maze = new Maze();
    public MazeBuilder AddRock(int x, int y)
    {
        return this;
    }
}
```

```
public MazeBuilder AddWater(int x, int y)
{
    return this;
}
```

// ещё много подобных методов AddXXX для добавления всего в maze.

// Метод, который создает финальный лабиринт и возвращает его.

```
public Maze Build()
{
    // ...
    return maze;
}
}
```

Теперь вы решили сделать MazeBuilder расширяемым, чтобы кто угодно мог унаследоваться от него и добавить новых методов. Примерно вот так:

```

public class ScaryMazeBuilder : MazeBuilder
{
// Хотим добавить новые возможности к MazeBuilder-у
public ScaryMazeBuilder AddGhost(int x, int y)
{
//...
return this;
}

// ещё много методов добавления страшилок в лабиринт.
}

```

Это почти сработало. Вот этот тест компилируется и работает:

```

public void TestBuilder_Ok()
{
var builder = new ScaryMazeBuilder();
var maze = builder
.AddGhost(5, 5)
.AddGhost(15, 5)
.AddRock(5, 6)
.AddRock(5, 4)
.AddRock(4, 5)
.Build();
}

```

Но вот если поменять порядок вызова методов, то компилироваться перестает:

```

public void TestBuilder_ShouldBeOk()
{
var builder = new ScaryMazeBuilder();
var maze = builder
.AddRock(5, 6)
.AddRock(5, 4)
.AddRock(4, 5)
// .AddGhost(5, 5) // не работает :(
// .AddGhost(15, 5) // не работает :(
.Build();
}

```

Доработайте класс `MazeBuilder` так, чтобы его можно было полноценно расширять. Покажите, что расширение работает, доработав класс `ScaryMazeBuilder`.

LMS-платформа – не предусмотрена

5.2.3. Домашняя работа № 1

Примерный перечень тем

1. Инкапсуляция

Примерные задания

Нейронная сеть состоит из нейронов, каждый из которых имеет вектор весов. Иногда нужно иметь доступ к весам отдельных нейронов, например, при их инициализации. Однако, при разработке алгоритмов обучения оказывается удобным работать со всеми весами в сети как с единым вектором. Таким образом, нужно организовать различные виды доступа к одним и тем же данным.

Скачайте проект `Incapsulation.Weights` и напишите класс `Indexer`, который создается как обертка над массивом `double[]`, и открывает доступ к его подмассиву некоторой длины, начиная с некоторого элемента. Ваше решение должно проходить тесты, содержащиеся в проекте. Как и всегда, вы должны следить за целостностью данных в `Indexer`.

LMS-платформа – не предусмотрена

5.2.4. Домашняя работа № 2

Примерный перечень тем

1. Делегирование

Примерные задания

Часто делегаты можно использовать для тонкой настройки алгоритмов, что позволит использовать один и тот же код для решения несколько разных задач.

Скачайте проект `Delegates.TreeTraversal`

Перед вами три задачи:

Дано дерево категорий продуктов, в каждой категории могут быть другие категории и собственно продукты. Вам нужно вывести список продуктов.

Дано дерево задач, каждая задача может содержать подзадачи. Вам нужно вывести список таких задач, у которых нет подзадач.

Дано бинарное дерево, у которого каждый узел содержит значение. Вам нужно вывести все значения в листьях этого дерева.

Вам нужно написать один алгоритм обхода дерева, который бы принимал в качестве аргументов делегаты, объясняющие алгоритму, как обходить дерево и какие величины выводить.

LMS-платформа – не предусмотрена

5.2.5. Домашняя работа № 3

Примерный перечень тем

1. DDD

Примерные задания

Скачайте проект `Ddd.Taxi`.

Все Value-типы, согласно DDD, должны поддерживать семантику значений, то есть сравниваться по содержимому своих свойств. Каждый раз реализовывать `Equals`, `GetHashCode` и `ToString` соответствующим образом — довольно муторное занятие. Часто для этого создают базовый класс, наследование от которого реализует нужным образом все эти стандартные методы. Это вам и предстоит сделать!

В рамках этого задания сравнивать Value-типы можно только по значению их публичных свойств, без учета значения полей. Хотя как правильно это стоит делать на практике — вопрос дискуссионный и, скорее, предмет договорённостей в вашей команде.

В файле `Infrastructure/ValueType.cs` реализуйте класс `ValueType` так, чтобы проходили все тесты в файле `ValueType_Tests.cs`.

LMS-платформа – не предусмотрена

5.2.6. Домашняя работа № 4

Примерный перечень тем

1. Модульность

Примерные задания

Скачайте проект `SRP.ControlDigit`.

В серийные номера, номера счетов и коды продуктов обычно включают так называемый контрольный разряд, значение которого вычисляется по остальным цифрам номера. Он нужен, чтобы подтверждать отсутствие ошибок при вводе этих номеров вручную или при считывании их с помощью сканеров.

Есть несколько стандартизированных алгоритмов вычисления контрольного разряда. Прочитать их краткое описание с примерами можно в соответствующей статье википедии.

В проекте уже реализован один метод `UPC`. Причем реализован без какой-либо декомпозиции. Вам нужно реализовать две других функции, но это не главное. Главное, провести декомпозицию имеющегося кода, разбив его на небольшие повторно используемые функции.

Весь код логически разделите на общий код и специфичный.

Общий код поместите в класс `Extensions` в виде методов расширения. Там должны оказаться только методы с понятными логичными сигнатурами, которые имеют шанс быть использованными в другом контексте.

Специфичный код — это код, который использует специфику задачи, например содержит название алгоритма в имени метода. Весь специфичный код оставьте в классе `ControlDigitAlgo` в виде приватных методов (ведь мы все равно не ожидаем, что они будут повторно использованы каким-то кодом извне).

Постарайтесь минимизировать количество специфичного кода и максимизировать количество общего кода.

Считайте, что эти методы не будут вызываться слишком часто, поэтому не нужно пытаться их оптимизировать, вместо этого сосредоточьтесь на простоте, понятности и возможности повторного использования.

В этой задаче действует ограничение на длину методов — не более 10 строк. Но не надо пытаться записывать сразу много операторов в одной строке — это дурной тон в программировании. Вместо этого постарайтесь выделить из длинных методов самостоятельные примитивы в отдельные методы.

LMS-платформа – не предусмотрена

5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля

5.3.1. Зачет

Список примерных вопросов

1. Инкапсуляция
 2. Наследование и полиморфизм
 3. Generics
 4. Делегирование
 5. Рефлексия
 6. DDD
 7. FluentAPI
 8. Модульность
 9. Управление зависимостями
 10. DI-контейнеры
 11. Функциональный стиль
 12. Управление ресурсами
 13. Работа с файлами
 14. Работа с файлами
- LMS-платформа – не предусмотрена

5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности

Направление воспитательной деятельности	Вид воспитательной деятельности	Технология воспитательной деятельности	Компетенция	Результаты обучения	Контрольно-оценочные мероприятия
Профессиональное воспитание	профориентационная деятельность	Технология формирования уверенности и готовности к самостоятельной успешной профессиональной деятельности	ОПК-5	У-2 П-2	Домашняя работа № 1 Домашняя работа № 2 Домашняя работа № 3 Домашняя работа № 4 Зачет Контрольная работа № 1 Контрольная работа № 2 Лабораторные занятия Лекции