

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ  
**ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР**  
**«КОЛЬСКИЙ НАУЧНЫЙ ЦЕНТР РОССИЙСКОЙ АКАДЕМИИ НАУК»**  
(ФИЦ КНЦ РАН)

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНЫХ РАБОТ**

По дисциплине Б1.О.09 Современные технологии программирования

указывается цикл (раздел) ОП, к которому относится дисциплина, название дисциплины

для направления подготовки (специальности) 09.04.02 Информационные системы и технологии

код и наименование направления подготовки (специальности)

направленность программы (профиль) Информационные системы предприятий и учреждений

наименование профиля /специализаций/образовательной программы

**Квалификация выпускника, уровень подготовки**

Магистр

(указывается квалификация (степень) выпускника в соответствии с ФГОС ВО)

**Апатиты**

**2020**

**Лист согласования**

1 Разработчик:

доцент  
должность

УАиМ

  
подпись

Н.А. Тоичкин  
И.О. Фамилия

2 Методические указания рассмотрены и одобрены на заседании учебно-методической комиссии управления аспирантуры и магистратуры 29 июня 2020 г., протокол № 02.

Председатель УМК УАиМ

29.06.2020  
дата

подпись



Л.Д. Кириллова  
И.О.Фамилия

## Пояснительная записка

1. **Методические указания** составлены в соответствии с требованиями федерального государственного образовательного стандарта по образовательной программе высшего образования – программе магистратуры по направлению подготовки 09.04.02 Информационные системы и технологии, утвержденного приказом Минобрнауки России от 19.09.2017 № 917.

2. **Цель дисциплины (модуля)** формирование у обучающихся целостного представления о современных подходах к созданию программных продуктов; технологиях, методах и инструментальных средств разработки, модификации и сопровождения программных комплексов и систем.

### **Задачи дисциплины:**

- получить навыки использования современных техник программирования разработки прикладного программного обеспечения;
- изучить работу с разнообразными структурами данных;
- научиться готовить тестовые задания для отладки отдельных программных модулей и программного комплекса в целом.

3. **Требования к уровню подготовки обучающегося в рамках данной дисциплины.**

Процесс изучения дисциплины (модуля) «Современные технологии программирования» направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО 09.04.02 Информационные системы и технологии (уровень магистратуры), представленных в таблице 1.

**Таблица 1 – Компетенции, формируемые в процессе изучения дисциплины «Современные технологии программирования»**

№ п/п	Код компетенции	Содержание компетенции
1.	ОПК-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач.

4. **Планируемые результаты обучения по дисциплине (модулю) «Современные технологии программирования».**

Результаты формирования компетенций и обучения представлены в таблице 2.

**Таблица 2 – Планируемые результаты обучения**

№ п/п	Код компетенции	Компоненты компетенции, степень их реализации	Результаты обучения
1.	ОПК-2	Компоненты компетенции со-	<i>Знать:</i>

		<p>относятся с содержанием дисциплины и компетенция реализуется полностью</p>	<ul style="list-style-type: none"> <li>– современные концепции программирования: структурный подход, объектно-ориентированный подход, функциональный подходи их применение в языке C#;</li> <li>– технологии обобщенного программирования;</li> <li>– основные структуры данных и их применение при разработке различных алгоритмов;</li> <li>– задачи разработки прикладного программного обеспечения.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>– разрабатывать иерархические схемы программ;</li> <li>– использовать механизм организации запросов к источнику данных LINQ для написания выразительного декларативного кода;</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>– методами разработки приложений с использованием возможностей современных технологий программирования;</li> <li>– навыками работы в современных средах разработки.</li> </ul>
--	--	---	--

## ПЕРЕЧЕНЬ РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

### Основная литература:

1. Павловская Т.А., Программирование на языке высокого уровня C# / Павловская Т.А. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : [https://www.studentlibrary.ru/book/intuit\\_281.html](https://www.studentlibrary.ru/book/intuit_281.html) (дата обращения: 06.12.2020).
2. Чеповский А.М., Common Intermediate Language и системное программирование в Microsoft .NET / Чеповский А.М., Макаров А.В., Скоробогатов С.Ю. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. (Основы информатики и математики) - ISBN 5-94774-410-4. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN5947744104.html> (дата обращения: 06.12.2020).

3. Лекции по курсу. Электронное изд. <https://moodle.arcticsu.ru/course/view.php?id=159>

**Дополнительная литература:**

4. Седжвик Р. Алгоритмы на С++. М.: Национальный Открытый Университет «ИНТУИТ». Режим доступа: [https://biblioclub.ru/index.php?page=book\\_red&id=429164&sr=1](https://biblioclub.ru/index.php?page=book_red&id=429164&sr=1)
5. Тоичкин Н.А., Козлова Ю.Г., Богатилов В.Н. Паттерны проектирования: учеб.-метод. пособие по выполнению лаб. работ, (учебное пособие)/ Н.А. Тоичкин, В.Н. Богатилов, Ю.Г. Козлова. Тверь: ТвГТУ, 2015. 48 с. Электронное изд. <https://moodle.arcticsu.ru/course/view.php?id=159>

## СОДЕРЖАНИЕ ПРОГРАММЫ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

### Тема 1. Обзор современных технологий программирования.

Основные принципы структурного и модульного программирования. Иерархическое упорядочивание. Принципы тестирования. Модульное тестирование. Объектно-ориентированное программирование (ООП). Краткий обзор современных ООП языков программирования. Инструментальные среды разработки.

Рекомендуемая литература: [3], [5].

Вопросы для самоконтроля знаний:

1. Какие вызовы в программировании привели к созданию ООП.
2. Перечислите основные принципы ООП?
3. Что такое абстракция в терминах ООП?
4. В чем отличие класса от интерфейса?
5. Какие современные языки программирования вы знаете?
6. Каким образом осуществляется модульное тестирование?
7. Что такое паттерны проектирования и где они применяются.
8. Перечислите известные вам паттерны проектирования.

### Тема 2. Структуры данных, обобщенное программирование и перечисление коллекций.

Стеки, очереди, листы и словари. Обобщенное программирование. Определение и использование шаблона класса. Параметры шаблонов. Хэширование в словарях. Перечисление коллекций. Интерфейсы IEnumerable и IEnumerator и их реализация. Yield return.

Рекомендуемая литература: [1], [2], [3].

Вопросы для самоконтроля знаний:

1. Что такое стек? Перечислите операции со стеком.
2. Что такое очередь? Перечислите операции с очередью.
3. Дайте определение интерфейса.
4. Для чего используются интерфейсы IEnumerable и IEnumerator?
5. В чем смысл применения оператора Yield Return?
6. Как организовать перечисление коллекций с использованием foreach?

### **Тема 3. Введение в функциональное программирование в языке программирования C#: делегаты, лямбда – выражения.**

Делегаты. Постановка проблемы. Делегат как тип данных. Делегат на картах памяти. Дженерик-делегаты. Func и Action. Анонимные делегаты. Лямбда-выражение. Примеры лямбд. Замыкание: как работает замыкание, пример на картах памяти, ловушка замыкания.

Рекомендуемая литература: [1], [2], [3].

Вопросы для самоконтроля знаний:

1. Что такое делегат?
2. Что такое анонимный делегат?
3. Где хранится делегат и ссылка на него?
4. Для чего применяются лямбда выражения, примеры использования.
5. Как используются Func и Action? Приведите пример их использования.
6. Примеры использования лямбда выражений в тестах.
7. Что такое замыкание и ловушка замыкания.

### **Тема 4. Механизм запросов для коллекций объектов в памяти.**

LINQ. LINQ to Objects. Методы Where, Select, ToList, SelectMany в LINQ. Методы фильтрации и преобразования. Method chaining. Функции агрегирования и группировки.

Рекомендуемая литература: [1], [2], [3].

Вопросы для самоконтроля знаний:

1. Что LINQ?
2. Что такое ленивый метод?
3. Какие из представленных методов являются ленивыми: Where, Select, ToList?
4. В чем особенности последовательности вызовов ленивых и обычных методов?
5. Перечислите методы фильтрации и преобразования в LINQ, приведите примеры работы с ними.
6. Перечислите методы агрегирования и группировки в LINQ, приведите примеры работы с ними.
7. Для чего в LINQ используются методы ToDictionary и ToLookup?
8. В чем состоит отличие Select и SelectMany. Приведите примеры использования.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

Итоговый уровень знаний обучающихся, приобретенный при изучении дисциплины «Современные технологии программирования», проверяется на экзамене.

1. Основные принципы структурного и модульного программирования. Иерархическое упорядочивание.
2. Принципы тестирования. Модульное тестирование.
3. Объектно-ориентированное программирование (ООП). Краткий обзор современных ООП языков программирования. Инструментальные среды разработки.
4. Стек. Стеки для анализа скобочных выражений. Стеки для вычисления (обратная польская запись).
5. Очередь. Очередь на связных списках. Универсальная очередь и даункасты. Реализация очереди при помощи дженерик-параметра.
6. Очередь для скользящего среднего.
7. Дженерики и сортировка массивов.
8. Проблема возвращения нескольких значений из метода.
9. `foreach`, `IEnumerable` и `IEnumerator`. Сущности интерфейсов `IEnumerable` и `IEnumerator`. Реализация интерфейса `IEnumerator`.
10. Реализация интерфейса `IEnumerator`.
11. `Yield return`.
12. Ленивые коллекции. Задача генерации бесконечных последовательностей.
13. `yield return` в рекурсивных методах. Примеры: перебор всех подмножеств, множества, разбиение множества.
14. Листы и индексация. Метод `Contains`. Метод `Equals`.
15. Листы и индексация. Перегрузка операторов.
16. Хеширующие функции. Примеры.
17. Использование хэшей в алгоритме поиска подстроки в строке.
18. Класс `Dictionary`. `GetHashCode`.
19. Делегаты. Постановка проблемы. Делегат как тип данных.
20. Делегат на картах памяти. Дженерик-делегаты.
21. `Func` и `Action`. Анонимные делегаты. Лямбда-выражение. Примеры лямбд.
22. Замыкание: как работает замыкание, пример на картах памяти, ловушка замыкания (пример).
23. Делегаты для диагностики кода (на примере сортировок). Делегаты в разборе арифметических выражений. Делегаты в вычислении производной.
24. Лямбда-выражения в тестах.
25. LINQ. LINQ to Objects. Метод `Where` в LINQ: реализация метода `Where`. Метод `Select` в LINQ: реализация метода `Select`. Метод `ToList` в языке LINQ.
26. Последовательность вызовов `Where` и `Select`. Последовательность вызовов с `ToList`. Особенности последовательности вызовов ленивых и обычных методов.
27. Фильтрация и преобразование. Методы фильтрации и преобразования. `Take`, `Skip`, `ToArray`, `ToList`.
28. Method chaining. `SelectMany`. `OrderBy` и `Distinct`.
29. Функции агрегирования. Группировка.
30. `ToDictionary` и `ToLookup`.

Рекомендуемая литература: [1], [2], [3], [4], [5].