

МИНОБРНАУКИ РОССИИ

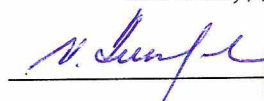
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
«КОЛЬСКИЙ НАУЧНЫЙ ЦЕНТР РОССИЙСКОЙ АКАДЕМИИ НАУК»
(ФИЦ КНЦ РАН)

УТВЕРЖДАЮ

Начальник управления аспирантуры и магистратуры

ФИЦ КНЦ РАН

к.г.-м.н., доцент И.В. Чикирёв



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

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 Фонд оценочных средств рассмотрен и одобрен 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

Лист переутверждения

Фонд оценочных средств переутвержден на 2021/2022 учебный год без изменений и дополнений.

Председатель УМК УАиМ Л.Д. Кириллова Л.Д. Кириллова

Основание: протокол № 2 от «29» июня 2021 г.

Фонд оценочных средств переутвержден на _____ / _____ учебный год без изменений и дополнений.

Председатель УМК УАиМ _____ Л.Д. Кириллова

Основание: протокол № _____ от « _____ » _____ 20 _____ г

Фонд оценочных средств переутвержден на _____ / _____ учебный год без изменений и дополнений.

Председатель УМК УАиМ _____ Л.Д. Кириллова

Основание: протокол № _____ от « _____ » _____ 20 _____ г

Фонд оценочных средств переутвержден на _____ / _____ учебный год без изменений и дополнений.

Председатель УМК УАиМ _____ Л.Д. Кириллова

Основание: протокол № _____ от « _____ » _____ 20 _____ г

Фонд оценочных средств переутвержден на _____ / _____ учебный год без изменений и дополнений.

Председатель УМК УАиМ _____ Л.Д. Кириллова

Основание: протокол № _____ от « _____ » _____ 20 _____ г

Лист изменений, вносимых в ФОС по дисциплине «Современные технологии программирования»

В фонд оценочных средств вносятся следующие изменения и дополнения:

1. _____

2. _____

3. _____

Дополнения и изменения внесены и одобрены на заседании учебно-методической комиссии управления аспирантуры и магистратуры ФИЦ КНЦ РАН

от «____» _____ г., протокол № _____.

Председатель УМК УАиМ _____ Л.Д. Кириллова

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ
АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Общие сведения

1.	Отдел	Аспирантуры и магистратуры
2.	Направление подготовки	09.04.02 Информационные системы и технологии Профиль – «Информационные системы предприятий и учреждений»
3.	Дисциплина (модуль)	Б1.О.09 Современные технологии программирования
4.	Количество этапов формирования компетенций (ДЕ, разделов, тем и т.д.)	4

Перечень компетенций:

- Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач (ОПК-2).

Критерии и показатели оценивания компетенций на различных этапах их формирования

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
1. Обзор современных технологий программирования.	ОПК-2	современные концепции технологий программирования; концепцию и составные части структурного и объектно-ориентированного программирования; принципы применения шаблонов классов	выполнять кодирование, отладку и тестирование отдельных программных модулей и программного комплекса в целом	методами решения задач с использованием возможностей современных технологий программирования	Раздел вопросов контрольной работы, проверка практической работы
2. Структуры данных и обобщенное программирование.	ОПК-2	способы применения различных структур данных в процессе разработки программного обеспечения	исследовать методы и способы решения задач по разработке программных комплексов; составлять план разработки программных модулей	методами решения задач с использованием возможностей различных структур данных и методов обобщенного программирования	
3. Введение в функциональное программирование в языке программирования C#: делегаты, лямбда – выражения.	ОПК-2	понятия: делегатов, анонимных делегатов и лямбда выражений	применять технологии функционального программирования для решения конкретных прикладных задач	навыками использования лямбда выражений	Раздел вопросов контрольной работы, проверка практической работы
4. Механизм запросов для коллекций объектов в памяти.	ОПК-2	синтаксис организации запросов к коллекциям находящимся в памяти	использовать механизм организации запросов к источнику данных LINQ для написания выразительного декларативного кода	навыками использования LINQ	Раздел вопросов контрольной работы, проверка практической работы

Критерии и шкалы оценивания

1. Выполнение практической работы

15 баллов выставляется, студент выполнил полностью все задания указанные в лабораторной работе и может аргументировано пояснить ход своего решения.

13 баллов выставляется, если студент выполнил не менее 85 % заданий указанных в лабораторной работе, и может аргументировано пояснить ход своего решения и указать.

12 баллов выставляется, если студент решил не менее 50% заданий указанных в лабораторной работе, и может аргументировано пояснить ход своего решения.

0 баллов выставляется, если студент не может аргументированно пояснить ход своего решения.

В случае если сроки сдачи работ превышены, количество баллов сокращается на 5 баллов.

Темы практических занятий, описание структуры и содержания, вопросы для подготовки к практическому занятию представлены в методических указаниях к практическим занятиям по дисциплине.

2. Экзамен

Оценка	Критерии оценивания
«отлично» (20 баллов)	Ставится в том случае, когда обучающийся обнаруживает систематическое и глубокое знание программного материала по дисциплине, умеет свободно ориентироваться в вопросе. Ответ полный и правильный на основании изученного материала. Выдвинутые положения аргументированы и иллюстрированы примерами. Материал изложен в определенной логической последовательности, осознанно, литературным языком, с использованием современных научных терминов; ответ самостоятельный. Обучающийся уверенно отвечает на дополнительные вопросы. Дисциплинарные компетенции сформированы на высоком уровне.
«хорошо» (15 баллов)	Ставится в том случае, когда обучающийся обнаруживает полное знание учебного материала, демонстрирует систематический характер знаний по дисциплине. Ответ полный и правильный, подтвержден примерами; но их обоснование не аргументировано, отсутствует собственная точка зрения. Материал изложен в определенной логической последовательности, при этом допущены 2-3 несущественные погрешности, исправленные по требованию экзаменатора. Обучающийся испытывает незначительные трудности в ответах на дополнительные вопросы. Материал изложен осознанно, самостоятельно, с использованием современных научных терминов, литературным языком. Дисциплинарные компетенции сформированы на продвинутом уровне.
«удовлетворительно» (10 баллов)	Ставится в том случае, когда обучающийся обнаруживает знание основного программного материала по дисциплине, но допускает погрешности в ответе. Ответ недостаточно логически выстроен, самостоятелен. Основные понятия употреблены правильно, но обнаруживается не-

	<p>достаточное раскрытие теоретического материала. Выдвигаемые положения недостаточно аргументированы и не подтверждены примерами; ответ носит преимущественно описательный характер. Обучающийся испытывает трудности в ответах на вопросы. Научная терминология используется недостаточно.</p> <p>Дисциплинарные компетенции сформированы на базовом уровне.</p>
«неудовлетворительно» (0 баллов)	<p>Выставляется обучающемуся, обнаружившему пробелы в знаниях основного учебного материала по дисциплине. При ответе обнаружено непонимание обучающимся основного содержания теоретического материала или допущен ряд существенных ошибок, которые обучающийся не может исправить при наводящих вопросах экзаменатора, затрудняется в ответах на вопросы. Ответ носит поверхностный характер; наблюдаются ошибки в использовании научной терминологии.</p> <p>Дисциплинарные компетенции не сформированы.</p>

3. Контрольная работа (тест)

Процент правильных ответов	41-60	61-80	81-100
Количество баллов	19	22	25

Пример типового тестового задания (контрольной работы)

1. Какая компонента платформы .NET отвечает за автоматическое обнаружение, загрузку и управление типами .NET ?

- + : Common Language Runtime.
- : Common Type System.
- : Common Language Specification.

2. Какая компонента платформы .NET отвечает за спецификации всех возможных типов данных и программных конструкций, поддерживаемых исполняющей средой?

- : Common Language Runtime.
- + : Common Type System.
- : Common Language Specification.

3. Укажите истинные утверждения.

- + : Сборки представляют собой самоописываемые сущности.
- + : Перед использованием CIL код должен обязательно компилироваться на лету.
- + : .NET использует оперативный компилятор в соответствии с целевым ЦП и оптимизирует его согласно лежащей в основе платформы.
- : Для исполнения управляемого кода, CIL-инструкции могут использоваться напрямую без компиляции.

4. Изучите код ниже:

```

1 class Sample1 {
2     static void F() {
3         i = 1;
4     }
5     static int i = 0;
6 }
```


Корректно ли использование поля класса ДО его определения как в примере выше?

+ : Да, корректно.

- : Нет, это не скомпилируется.

- : Это скомпилируется, но при выполнении метода F возникнет ошибка.

5. Какая компонента платформы .NET отвечает за автоматическое обнаружение, загрузку и управление типами .NET ?

+ : Common Language Runtime

- : Common Type System

- : Common Language Specification

6. Какая компонента платформы .NET отвечает за спецификации всех возможных типов данных и программных конструкций, поддерживаемых исполняющей средой?

- : Common Language Runtime

+ : Common Type System

- : Common Language Specification

7. Укажите истинные утверждения.

- + : Сборки представляют собой самоописываемые сущности.
- + : Перед использованием CIL код должен обязательно компилироваться на лету.
- + : .NET использует оперативный компилятор в соответствии с целевым ЦП и оптимизирует его согласно лежащей в основе платформы.
- : Для исполнения управляемого кода, CIL-инструкции могут использоваться напрямую без компиляции.

8. Изучите код ниже:

```
1 class Sample1 {
2     static void F() {
3         i = 1;
4     }
5     static int i = 0;
6 }
```

Корректно ли использование поля класса ДО его определения как в примере выше?

- + : Да, корректно
- : Нет, это не скомпилируется
- : Это скомпилируется, но при выполнении метода F возникнет ошибка

9. Что будет выведено при вызове метода Mixed()?

```
1 class Sample3 {
2     static string who = "class";
3
4     static void Mixed() {
5         Console.Write(who + " ");
6         string who = "Mixed";
7         Console.Write(who);
8     }
9 }
```

- : class class
- : class Mixed
- : Mixed Mixed
- : Mixed class
- + : Ничего. Будет ошибка компиляции

10: Как можно дополнить код, чтобы он начал компилироваться? Выберите все возможные варианты.

```
1 namespace Slide01
2 {
3     class Program
4     {
5         static void Main()
6         {
7             System.Console.WriteLine(Min(4, 2, 3));
8         }
9
10        static int Min(int a, int b, int c)
11        {
12            return Math.Min(a, Math.Min(b, c));
13        }
14    }
15 }
```

- : Дописать "using System.Console;" в начало
- +: Написать "System.Math.Min" вместо "Math.Min"
- : Это скомпилируется, но при выполнении метода возникнет ошибка
- : Дописать "using System.Math;" в начало
- +: Дописать "using System;" в начало

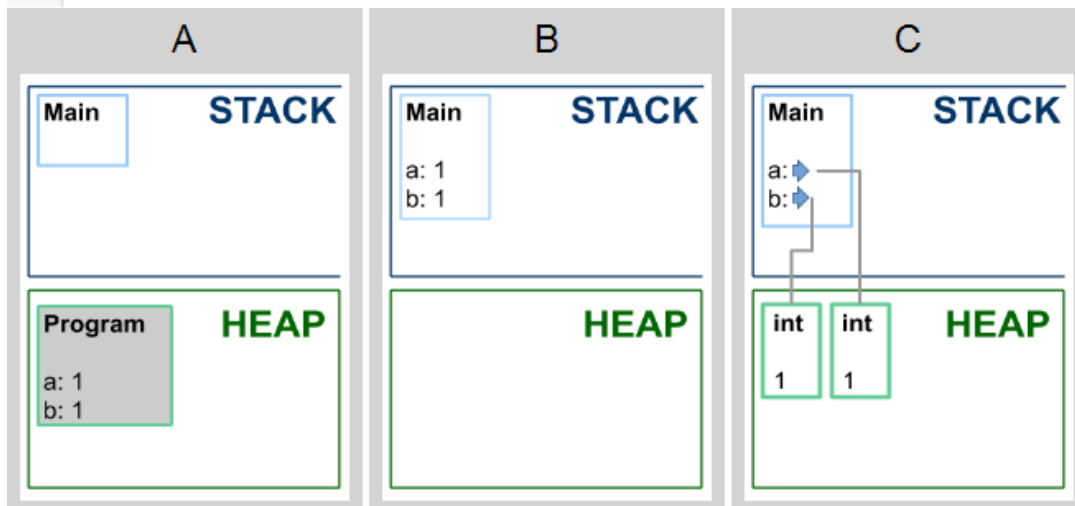
11. Что стилистически неверно в коде выше? Выберите все верные варианты.

```
1 namespace StyleErrors
2 {
3     class Program
4     {
5         static public void Main()
6         {
7             var myinteger = "sotnya";
8         }
9     }
10 }
```

- +: Название переменной не соответствует содержанию
- : void в сигнатуре Main() не нужен
- : Незачем создавать класс, если всего один метод внутри
- +: Название переменной не соответствует правилу camelCase

12. Какая из карт памяти соответствует коду к моменту выхода из метода Main?

```
1 public class Program
2 {
3     public static void Main()
4     {
5         int a = 1;
6         int b = 1;
7         // ?
8     }
9 }
```



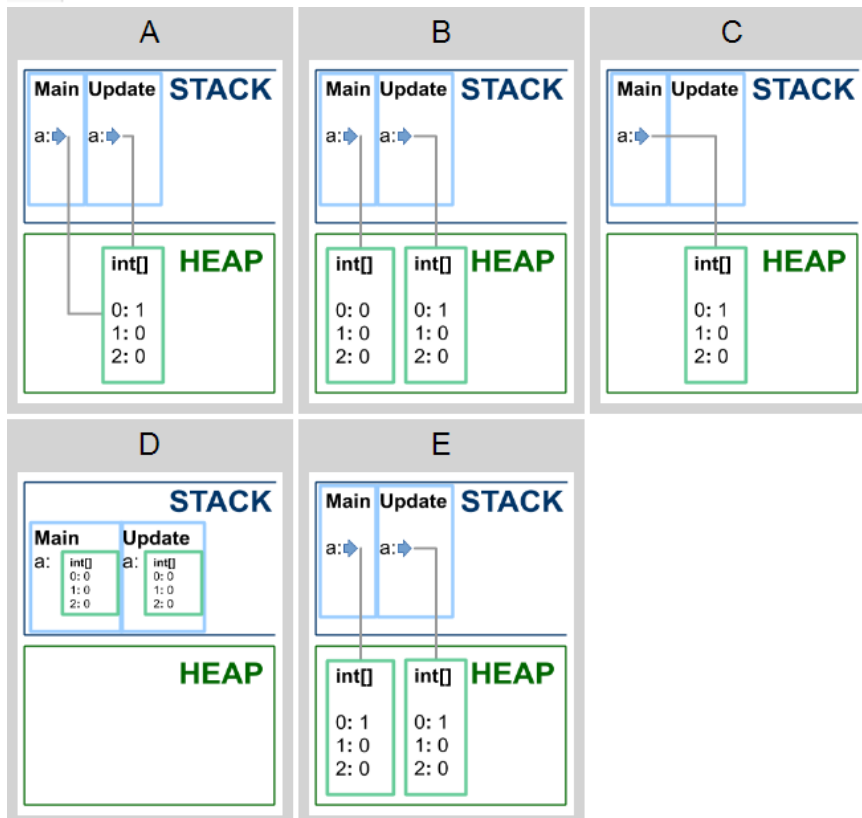
- : A
- +: B
- : C

13. Какая из карт памяти соответствует коду к моменту выхода из метода Update?

```

1 public class Program
2 {
3     static void Update(int[] a)
4     {
5         a = new int[3];
6         a[0]++;
7         // ?
8     }
9
10    static void Main()
11    {
12        int[] a = new int[3];
13        Update(a);
14    }
15 }

```



- : A
- +: B
- : C
- : D
- : E

14. Массивы отлично использовать для...

- +: хранения набора однотипных данных
- : хранения набора разнотипных данных
- : хранения разных свойств одного объекта
- +: поиска значения по его номеру

15. Отметьте все верные факты про массивы a и b

```
var a = new int[2, 3];
var b = new int[2] [];
```

- +: Все ячейки массива `a` содержат значение 0
- : Все ячейки массива `a` содержат значение null
- : Все ячейки массива `b` содержат значение 0
- +: Все ячейки массива `b` содержат значение null
- +: Массив `a` двумерный
- : Массив `b` двумерный
- +: Тип элементов `a` — int
- +: Тип элементов `b` — int[]

16. Какое утверждение является верным для ситуации передачи в метод, ссылочного типа по значению?

- +: вызываемый код может изменять значения данных состояния объекта, но не сам объект, на который указывает входная ссылка.
- : вызываемый код может изменять значения данных состояния объекта, а также сам объект, на который указывает входная ссылка.

17. Отметьте все верные факты про кодировки

- +: Кодировка — это способ преобразования символов в байты и обратно
- : Кодировка — это только лишь сопоставление номера каждому символу
- : В кодировке UTF-8 все символы кодируются более, чем одним байтом
- +: Формат записи латинских букв в кодировке UTF-8 совпадает с форматом в старой кодировке ASCII
- : Кодировка — это непонятная и вредная штука, она портит все буквы, превращая их в кракозябры :(

18. Верна ли оценка $n = O(n)$?

- : Верно
- +: Неверно

19. Зачем нужно переносить часть кода в библиотеки — отдельные проекты, компилирующиеся в dll-файлы?

- : Это необходимо для написания модульных тестов
- +: Чтобы можно было повторно использовать код из библиотек в других проектах
- : Это ускоряет выполнение программы
- : Это ускоряет компиляцию

20. Почему стоит отделять в своем коде логику от ввода и вывода

- : Это незыблемое правило, которому нужно следовать всегда без обсуждений и раздумий
- : Тестировать программу через ввод-вывод невозможно
- +: Логику, отделенную от ввода-вывода проще тестировать
- +: Логика может пригодиться в других обстоятельствах, а если она будет объединена с вводом-выводом, ее не получится использовать повторно

21. Дополнительные плюсы наличия автоматических тестов
 +: Тратится меньше усилий на выявление и устранение ошибок после старта эксплуатации
 +: Рефакторинг становится менее опасной затеей
 -: Повышается скорость разработки
 +: Повышается доверие к коду у других программистов
22. Верна ли оценка $n = O(n^2)$?
 +: Верно
 -: Неверно
23. Алгоритм принимает на вход два числа N и M. Какой размер его входа согласно теории алгоритмов?
 -: N+M
 -: N×M
 -: $\log_{10} N + \log_{10} M$
 +: Зависит от выбранного алфавита и способа кодирования входа

Вопросы к экзамену по дисциплине «Современные технологии программирования»

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 IEnumerable.
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.