**Санкт-Петербургский филиал федерального государственного**

**автономного образовательного учреждения высшего образования**

**«Национальный исследовательский университет**

**"Высшая школа экономики"»**

Факультет Санкт-Петербургская школа

физико-математических и компьютерных наук

Департамент информатики

**Рабочая программа дисциплины**   
 Программирование на платформе .NET

для образовательной программы «Программирование и анализ данных»

направления подготовки 01.04.02 «Прикладная математика и информатика»

уровень магистратура

Разработчик: Литвинов Ю. В., ylitvinov@hse.ru

Утверждена Деканом факультета Санкт-Петербургская школа

физико-математических и компьютерных наук

«31» августа 2018  г.

А.В. Омельченко \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург, 2018

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.*

# Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает требования к образовательным результатам и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих дисциплину «Программирование на платформе .NET», учебных ассистентов и студентов направления 01.04.02 «Прикладная математика и информатика» подготовки магистра, обучающихся по магистерской программе «Программирование и анализ данных» и изучающих дисциплину «Программирование на платформе .NET».

Рабочая программа дисциплины разработана в соответствии с:

* Образовательным стандартом НИУ ВШЭ по направлению подготовки 01.04.02 «Прикладная математика и информатика» (уровень магистратуры), утвержденным ученым советом Национального исследовательского университета «Высшая школа экономики», протокол от 03.03.2017 №02.
* Основной профессиональной образовательной программой «Программирование и анализ данных» направления подготовки 01.04.02 «Прикладная математика и информатика»;
* Объединенным учебным планом университета по образовательной программе «Прикладная математика и информатика», утвержденным в 2018 г.

# Цели освоения дисциплины

Целью освоения дисциплины «Программирование на платформе .NET» является формирование у студентов теоретических знаний и практических навыков программирования на платформе .NET и языках программирования C# и F#.

В результате изучения этой дисциплины студенты будут владеть основными концепциями .NET и языка C#, иметь практические навыки в создании объектно-ориентированных приложений, приложений с пользовательским интерфейсом, сетевых и многопоточных приложений на платформе .NET, владеть основными принципами функционального программирования на языке F#.

# Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

* + знать основные конструкции и идиомы языка C#, необходимые для изучения других дисциплин, предусмотренных базовым и рабочим учебными планами, а также для применения в профессиональной деятельности;
  + знать основные концепции платформы .NET, такие как исполнение на виртуальной машине, сборка мусора, многопоточность в .NET, работа с графическими пользовательскими интерфейсами в .NET;
  + уметь создавать программы, решающие задачи на языках С# и F#;
  + иметь навыки (приобрести опыт) использования сопутствующих инструментов разработки – отладчика, библиотек модульного тестирования, инструментов разработки пользовательских интерфейсов;
  + иметь навыки (приобрести опыт) в применении широкоиспользуемых технологий на платформе .NET для применения в профессиональной деятельности;

1. В результате освоения дисциплины студент осваивает следующие компетенции:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Компетенция** | **Код по ОС НИУ ВШЭ** | **Уровень формирования компетенции** | **Дескрипторы – основные признаки освоения (показатели достижения результата)** | **Формы и методы обучения, способствующие формированию и развитию компетенции** | **Форма контроля уровня сформированности компетенции** |
| Способен принимать управленческие решения и готов нести за них ответственность. | УК-5 | РБ  СД  МЦ | Знает основные принципы командной работы в IT сфере.  Производит совместную разработку проекта с учетом особенностей платформы .NET.  Владеет программными средствами командно-проектной деятельности, использует их для решения практических задач | Подготовка к практическим занятиям, работа на практических занятиях, групповая самостоятельная работа | Домашние задания |
| Способен анализировать, верифицировать, оценивать полноту информации в ходе профессиональной деятельности, при необходимости восполнять и синтезировать недостающую информацию. | УК-6 | РБ  СД  МЦ | Знает области целесообразного применения платформыы .NET, её преимущества и недостатки.  Читает свой и чужой код, проводит отладку программы.  Определяет целесообразность применения платформы .NET, языков C# и F# и сторонних библиотек для выбранной задачи. | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа | Домашние задания, устный  экзамен |
| Способен правильно использовать существующие и вводить новые понятия в области математики и информатики, интегрируя известные факты, концепции, принципы и теории, связанные с прикладной математикой и информатикой. | ОПК-2 | РБ  СД  МЦ | Формулирует алгоритм решения задачи в виде последовательности действий, выражаемой на языках F# или C#.  Реализует алгоритмы решения выбранной задачи на языках F# или C#.  Оценивает сложность работы алгоритма. | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа | Домашние задания, устный  экзамен |
| Способен обоснованно выбирать и применять в профессиональной деятельности современные компьютерные технологии в соответствии со спецификой решаемых задач, включая операционные системы, сетевые технологии, языки программирования, языки манипулирования данными, электронные библиотеки, пакеты прикладных программ. | ОПК-3 | РБ  СД  МЦ | Знает основные способы отладки и профилировки программ, разработанных на платформе .NET.  Проводит профилирование и отладку.  Тестирует производительность и безопасность программ, разработанных на платформе .NET. | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа | Домашние задания, устный  экзамен |
| Способен осуществлять целенаправленный многокритериальный поиск информации о новейших научных и технологических достижениях в сети Интернет и в других источниках. | ПК-7 | РБ  СД  МЦ | Знает основные книги, научные журналы, форумы и другие ресурсы с актуальными новостями платформы .NET и библиотек для неё.  Использует разнообразные информационные ресурсы для выбора оптимального способа реализации задачи с учетом особенностей платформы .NET.  Использует материалы конференций, книги и другие материалы для анализа особенностей разработки и реализации алгоритмов решения практических задач на языке С#. | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа | Домашние задания, устный  экзамен |
| Способен реализовывать модели и алгоритмы прикладной математики в виде компьютерных программ. | ПК-10 | РБ  СД  МЦ | Знает основные языковые конструкции С# и F#; способы организации программ на языках С# и F#; средства разработки (компиляторы, IDE, отладчики и т.д.).  Разрабатывает приложения для выбранной задачи на языках С# и F#.  Проектирует программы с использованием функционального стиля программирования, ООП и событийно-ориентированного программирования. | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа | Домашние задания, устный  экзамен |
| Способен оценивать корректность и воспроизводимость применения методов прикладной математики и информатики | ПК-11 | РБ  СД  МЦ | Знает основные этапы создания проектов с использованием платформы .NET.  Оценивает время, необходимое для проведения каждого этапа создания проекта  Использует навыки планирования разработки проекта на платформе .NET при решении практических задач. | Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа | Домашние задания, устный  экзамен |

# Место дисциплины в структуре образовательной программы

Для образовательной программы «Программирование и анализ данных» направления подготовки 01.04.02 «Прикладная математика и информатика» настоящая дисциплина относится к базовой профильной части блока дисциплин специализация "Разработка программного обеспечения".

Основные положения данной дисциплины используются для освоения следующих дисциплин:

* Управление разработкой ПО

# Тематический план учебной дисциплины

Курс рассчитан на 76 часа аудиторной нагрузки, из них 38 часов лекций и 38 часов практических занятий, общим объемом 4 зачетных единиц (152 часа).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Название раздела | Всего часов | Аудиторные часы | | | Самостоя- тельная работа |
| Лек- ции | Семи- нары | Практи- ческие занятия |
| 1 | Основы языка C# и платформы .NET | 25 | 6 | 0 | 6 | 12 |
| 2 | Разработка пользовательских интерфейсов с помощью библиотек Windows Forms и WPF | 25 | 6 | 0 | 6 | 12 |
| 3 | Многопоточное программирование | 25 | 6 | 0 | 6 | 12 |
| 4 | Углублённое изучение платформы .NET | 25 | 6 | 0 | 6 | 12 |
| 5 | Веб-программирование на платформе .NET | 25 | 6 | 0 | 6 | 12 |
| 6 | Функциональное программирование на языке F# | 27 | 8 | 0 | 8 | 16 |
| ИТОГО | | 152 | 38 | 0 | 38 | 76 |

1. **Содержание дисциплины**

|  |  |
| --- | --- |
| Раздел 1  Основы языка C# и платформы .NET. | |
| Тема 1 | Введение, C#.  Цель занятия – дать первоначальные знания и навыки работы с C#:  - краткое введение о платформе .NET и языке C#, смысл их существования и отличие от Java.  - Hello, world на C#.  - Объявление методов, ref- и out-параметры.  - Ссылочные типы и типы-значения, объявление классов и структур, разница с Java.  - Модификаторы видимости классов и членов класса, наследование, интерфейсы и абстрактные классы, base, преобразования типов.  - virtual, override, new method, partial, sealed.  - Разница вложенных классов в C# и в C++, Java.  - Особенности C# 7 и C# 8 – pattern matching, деконструкция, expression-bodied methods, switch expression |
| Тема 2 | Обзор платформы .NET.  Знакомство с библиотекой, виртуальной машиной (что именно делает, IL), CLI, CLR. Mono, .NET Core. NuGet. Компиляция-декомпиляция (ildasm, dotPeek). Запуск кода, верификация, binding redirect, appConfig/webConfig, безопасность (trust), AppDomains. Системы непрерывной интеграции на примере AppVeyor, среды разработки и инструменты. |
| Тема 3 | Исключения, модульное тестирование.  Исключения в .NET. Блоки try, catch, finally. Иерархия библиотечных классов-исключений, свойства Exception. Перебрасывание исключений, хорошие практики объявления исключений, особенности исключений в .NET. Рефлексия. Динамическая кодогенерация (на примере LINQ, Regexp). Модульное тестирование на C#: NUnit/MS Unit Testing Framework, Mock-объекты и NSubstitute. Хорошие практики написания юнит-тестов. |
| Тема 4 | Практическое занятие, модульное тестирование.  Написание в аудитории модульных тестов к заранее приготовленным классам с использованием библиотеки модульного тестирования NUnit, библиотеки mock-объектов NSubstitute |
| Тема 5 | Контейнеры и генерики.  Контейнеры, итераторы, стандартные контейнеры с особенностями их использования, генерики в .NET, написание своих генериков: параметрический полиморфизм, вариантность. LINQ. Статические конструкторы в генериках, конструкторы (инициализация) статических полей. SingleLinkedList/List/Array, Concurrent, using, IDisposable, disposable(bool), финализаторы.  Сравнение реализации генериков .NET и Java, сравнение с шаблонами в C++. |
| Тема 6 | Событийно-ориентированное программирование.  Делегаты, мультикастинг, библиотечные шаблоны Func/Action, ключевое слово event, анонимные методы, лямбда-выражения, замыкание, каноничное объявление события, ключевые слова add и remove. Библиотеки RX.NET, RX-LINQ. |
| Раздел 2  Разработка пользовательских интерфейсов с помощью библиотек Windows Forms и WPF | |
| Тема 1 | Пользовательский интерфейс, Windows Forms.  - рассказ про шаблон для WinForms-приложений в Visual Studio (а так же рассказ про особенности, не встречавшиеся ранее, например, регионы или partial-классы)  - события и обработчики  - библиотечные элементы управления (кратко)  - стандартные алгоритмы размещения элементов управления (Anchor, Dock, TableLayout)  - невизуальные элементы управления (на примере таймера) |
| Тема 2 | Практическое занятие, разработка приложения на Windows Forms.  В аудитории разрабатывается приложение с графическим интерфейсом: игра «Найди пару». |
| Тема 3 | Обзор библиотеки WPF.  Синтаксис и семантика языка XAML, структура классов WPF, логическое и визуальное дерево, понятия зависимых свойств и Routed events, концепция Data Binding, конвертеры. Геометрия элемента управления, управление положением элемента, преобразования систем координат. |
| Раздел 3  Многопоточное программирование. | |
| Тема 1 | Высокоуровневое многопоточное программирование.  Напоминание о понятии потока в операционных системах. Устройство потоков в ОС Windows, понятие планировщика. Класс Thread, пул потоков, класс Task, контекст исполнения. Отмена операций. Async/await, их использование в стандартной библиотеке. Task Parallel Library, PLINQ, потокобезопасные коллекции. |
| Тема 2 | Низкоуровневое многопоточное программирование.  Понятия гонки и взаимоблокировки. User-mode синхронизация: атомарные чтения/записи, Volatile, понятие Memory Fence. Interlocked, понятие lock-free, примеры. Команды управления планировщиком. Примитивы ядра ОС: WaitHandle, AutoResetEvent/ManualResetEvent, семафоры, мьютексы, гибридные конструкции (\*Slim). Монитор, ключевое слово lock, особенности его использования. |
| Раздел 4  Углублённое изучение платформы .NET. | |
| Тема 1 | Сборка мусора в .NET  Mark & Sweep-алгоритм сборки мусора, поколения в .NET, Large Object Heap, режимы и особенности настройки сборщика мусора, мониторинг. Финализаторы, SafeHandle, особенности IDisposable, внутреннее устройство финализации, ключевое слово fixed, WeakReference. |
| Раздел 5  Веб-программирование на платформе .NET | |
| Тема 1 | Обзор ASP.NET MVC Core, Entity Framework Core  Основные принципы работы веб-приложений и веб-сервисов, архитектура ASP.NET MVC Core, структура проекта, Hello, world. Язык описания правил генерации Razor. Валидация данных. Использование CSS и JavaScript для создания веб-фронтенда. Работа с базами данных на примере Entity Framework Core. |
| Тема 2 | Практическое занятие, разработка веб-приложения ASP.NET Core.  Совместная разработка приложения для регистрации на конференцию. |
| Раздел 6  Функциональное программирование на языке F# | |
| Тема 1 | F# как альтернативный язык разработки под .NET  Введение, общий обзор концепций F#. Основные конструкции: let, типы, конструкторы типов, кортежи. Подробно про функциональные типы. Рекурсия, взаимная рекурсия. Лямбда-функции. Операторы |> и >>. Частичное применение. matching. Ленивые типы данных, seq. Записи, discriminated union-ы. Хвостовая рекурсия, аккумулятор, Continuation Passing Style. |
| Тема 2 | Практическое занятие, функциональное программирование на F#  Пишем несколько несложных функций на F# вместе, в императивном и в функциональном стиле:  1. Посчитать факториал  2. Посчитать числа Фибоначчи (за линейное время)  3. Проверить, что все элемента списка различны  4. Реализовать функцию, возвращающую все элементы двоичного дерева, удовлетворяющие переданному как параметр условию  5. Описать тип «полином» и реализовать функцию, возводящую полином в заданную степень |
| Тема 3 | Типы и генерики в F#  Типы в F#. Автоматическое обобщение, стандартные генерик-функции, словари операций, преобразования типов, вывод типов, value restriction, point-free. ООП. units of measure (как средство иметь свою мини-систему типов, невидимую из C#) |
| Тема 4 | Вычислительные выражения в F#  Computation expressions. Workflow, Builder, монады, моноиды и эндоморфизмы. Async, операции внутри async, обработка исключений (Async.Catch), отмена операций |

1. **Оценочные средства**

**7.1. Формы контроля знаний студентов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип контроля | Форма контроля | 1 год | | | Параметры |
| 2 модуль | 3 модуль | 4 модуль |  |
| Текущий | Домашнее задание №1 | \* |  |  | Письменное домашнее задание |
| Домашнее задание №2 |  | \* |  | Письменное домашнее задание |
| Домашнее задание №3 |  |  | \* | Письменное домашнее задание |
| Итоговый | Устный экзамен |  |  | \* | Экзамен в устной форме |

**7.2. Критерии и шкалы оценки, примеры заданий**

**7.2.1. Текущий контроль**

## Текущий контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств.

## ДОМАШНЕЕ ЗАДАНИЕ №1

## *Домашнее задание №1 выдается студентам в одном варианте. Срок выполнения домашнего задания – 2 недели. Форма представления обучающимися домашнего задания – программа на языке С#.*

**Пример домашнего задания 1:**

Бор:

* Структура данных для эффективного хранения строк
* Представляет из себя дерево с символами на ребрах
* http://neerc.ifmo.ru/wiki/index.php?title=%D0%91%D0%BE%D1%80

Задача:

* Необходимо создать класс Trie, реализующий следующие методы:

/// Expected complexity: O(|element|)

/// Returns true if this set did not already contain the specified element

bool Add(string element);

/// Expected complexity: O(|element|)

bool Contains(string element);

/// Expected complexity: O(|element|)

/// Returns true if this set contained the specified element

bool Remove(string element);

/// Expected complexity: O(1)

int Size { get; set; }

/// Expected complexity: O(|prefix|)

int HowManyStartsWithPrefix(string prefix);

## Критерии оценивания и шкала оценки домашнего задания №1

|  |  |
| --- | --- |
| Оценка | Критерии выставления оценки |
| «Отлично»  (8-10) | Разработанная программа полностью удовлетворяет требованиям условия, не содержит ошибок реализации и оформления, имеются модульные тесты и комментарии. |
| «Хорошо»  (6-7) | Разработанная программа полностью удовлетворяет требованиям условия, однако имеет ошибки в реализации или грубые ошибки оформления, имеются модульные тесты и большинство требуемых комментариев. |
| «Удовлетворительно»  (4-5) | Предоставленная реализует только 3 или 4 из 5 требуемых методов или содержит грубые ошибки реализации, отсутствуют модульные тесты или комментарии, имеются ошибки в использовании языка или стандартной библиотеки. |
| «Неудовлетворительно»  (0-3) | Предоставленная программа работает некорректно; предоставленный код не компилируется; реализовано менее 3 из 5 требуемых методов, отсутствуют модульные тесты и комментарии. |

## ДОМАШНЕЕ ЗАДАНИЕ №2

## *Домашнее задание №2 выдается студентам в одном варианте. Срок выполнения домашнего задания – 2 недели. Форма представления обучающимися домашнего задания – программа на языке С#.*

## Пример домашнего задания №2:

## Реализовать свой собственный класс Option – абстракцию наличия или отсутствия значения. Должны присутствовать статические методы Some(T value) и None, создавать объекты Option должно быть можно только через них. Также должны быть методы IsSome, IsNone, Value (последний должен бросать исключение, если в объекте нет значения); метод Map(Func<T, U> f), возвращающий новый Option, равный None, если в исходном объекте не было значения, или Some(<результат применения f к значению внутри объекта>), если значение было (например, Option.Some(2).Map(x -> x \* 2) == Option.Some(4), Option.None().Map(x -> x \* 2) == Option.None()); метод Flatten(Option<Option<T>>), возвращающий "уплощённый" объект Option<T>.

## Критерии оценивания и шкала оценки домашнего задания №2

|  |  |
| --- | --- |
| Оценка | Критерии выставления оценки |
| «Отлично»  (8-10) | Программа полностью выполняет требования условия, имеются юнит-тесты на всю требуемую функциональность, комментарии, нет существенных ошибок оформления. |
| «Хорошо»  (6-7) | Имеются незначительные ошибки реализации либо серьёзные ошибки оформления, тесты не покрывают все требования условия. |
| «Удовлетворительно»  (4-5) | Имеются значительные ошибки реализации или программа не выполняет все требования условия, либо отсутствуют модульные тесты. |
| «Неудовлетворительно»  (0-3) | Программа не компилируется или выполняет требования условия менее, чем на 50%. |

## ДОМАШНЕЕ ЗАДАНИЕ №3

## *Домашнее задание №3 выдается студентам в одном варианте. Срок выполнения домашнего задания – 2 недели. Форма представления обучающимися домашнего задания – программа на языке С#.*

**Пример домашнего задания №3:**

Реализовать на WPF игру "крестики-нолики", позволяющую играть двум игрокам в режиме "hot seat" или с компьютером. При игре с компьютером должно быть можно выбрать один из двух уровней сложности. Должна отображаться статистика побед/поражений/ничьих, должно быть можно в любой момент начать новую партию.

## Критерии оценивания и шкала оценки домашнего задания №3

|  |  |
| --- | --- |
| Оценка | Критерии выставления оценки |
| «Отлично»  (8-10) | Реализовано графическое приложение, полностью удовлетворяющее требованиям условия, адекватно ведущее себя при изменении размера и использующее механизмы MVVM, Data Binding и команд WPF. Имеются комментарии к классам и основным методам. |
| «Хорошо»  (6-7) | Графическое приложение полностью удовлетворяет требованиям условия, но не в полной мере реализует рекомендуемые паттерны разработки приложений на WPF (не используются команды, Data Binding, код на C# сильно связан с XAML-разметкой). |
| «Удовлетворительно»  (4-5) | Реализованное приложение не в полной мере покрывает требования условия. |
| «Неудовлетворительно»  (0-3) | Реализованное приложение не работает либо имеет существенные дефекты, либо написано без использования библиотеки WPF. |

## 7.2.2. Итоговый контроль по дисциплине

Проверка качества освоения дисциплины производится в форме устного экзамена.

## УСТНЫЙ ЭКЗАМЕН

## *Устный экзамен проводится в форме ответов на вопросы экзаменационного билета и дополнительные вопросы по материалам курса. Экзаменационный билет содержит два вопроса. На подготовку ответа выделяется 40 минут.*

## Примерный перечень вопросов к экзамену:

## Язык C#, CLI, основы синтаксиса языка, ссылочные типы и типы-значения, преобразования типов, представление объектов в памяти

## Методы: способы передачи параметров, абстрактные, виртуальные и статические методы, модификаторы видимости.

## Платформа .NET: общее описание, CLR, IL, CTS

## Сборки: понятие сборки, сильные и слабые имена, загрузка сборки, GAC, Binding redirect, MsBuild

## NuGet, JIT, Ngen, понятие Managed Heap, AppDomain. Понятие целевой платформы, реализации: .NET Framework, Mono, .NET Core

## Исключения: бросание, перебрасывание и обработка, библиотечные исключения, свойства исключений, хорошие практики

## Рефлексия: загрузка сборки, создание экземпляра объекта, работа с полями и методами, dynamic

## Модульное тестирование: популярные библиотеки, хорошие практики, mock-объекты

## Контейнеры и генерики в .NET, энумераторы, открытые и закрытые типы, особенности статических полей в генериках, генерики и вложенные классы

## Генерики и наследование, ограничения на параметры-типы, ковариантность и контравариантность

## LINQ: основные методы, синтаксис, основные реализации, свои провайдеры

## Делегаты, их внутреннее устройство, delegate chaining, Invoke, шаблонные типы делегатов из стандартной библиотеки

## События, анонимные методы, лямбда-выражения, замыкания, каноничное объявление события, ручное управление подпиской

## Rx.NET, интерфейсы IObservable и IObserver, холодные и горячие последовательности, Rx.NET и LINQ, Subject

## WinForms: назначение, класс Control, обработка и валидация ввода, Data Binding, хорошие практики

## WPF: назначение и родственные технологии. XAML: атрибуты, конвертеры типов, расширения, коллекции. Структура классов WPF, логическое и визуальное дерево.

## WPF: зависимые свойства, routed events, команды. Data binding: конвертеры, направления привязки, валидация.

## WPF: стили, триггеры, шаблоны, ресурсы. Геометрия контрола, задание положения контрола и преобразования системы координат

## Потоки в .NET: классы Thread и ThreadPool, примитивы синхронизации уровня ядра: ключевое слово lock, мониторы, семафоры, WaitHandle, ManualResetEvent/AutoResetEvent, гибридные конструкции (\*Slim)

## Lock-free-программирование: основные понятия, атомарные чтения/записи, volatile, Interlocked, Compare-And-Swap

## Класс Task, исполнение и отмена асинхронных операций. Async/await.

## Сборка мусора, mark and sweep, поколения, Large Object Heap, когда происходит сборка мусора

## Режимы сборки мусора: Workstation/Server, многопоточная сборка. Динамическая настройка GC, ручное управление, мониторинг

## Финализаторы, IDisposable, using, реализация финализации, ручное управление жизнью объекта, fixed, WeakReference

## Веб-сервисы, веб-приложения, архитектура ASP.NET MVC. Работа с БД: понятие ORM, библиотека Entity Framework

## Язык F#: основные особенности, let-определения, кортежи, лямбды, списки, Option, взаимная рекурсия, pipe, композиция

## Каррирование, match, виды шаблонов, последовательности, записи, размеченные объединения

## Хвостовая рекурсия, паттерн “Аккумулятор”, Continuation Passing Style

## Генерики в F#, автоматическое обобщение, словари операций, касты, гибкие ограничения

## Методы отладки проблем типизации, value restriction, point-free, особенности арифметических операторов

## ООП в F#: методы, каррирование и кортежи при передаче параметров, конструкторы, свойства, мутабельность

## Модификаторы видимости, наследование, абстрактные классы и интерфейсы, реализация по умолчанию, объектные выражения, модули и пространства имён.

## Критерии оценивания и шкала оценки устного экзамена

|  |  |
| --- | --- |
| Оценка | Критерии выставления оценки |
| «Отлично»  (8-10) | Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. Продемонстрированы знания основных концепций по двум дополнительным вопросам. |
| «Хорошо»  (6-7) | Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы. Были даны в целом удовлетворительные ответы на два дополнительных вопроса. |
| «Удовлетворительно»  (4-5) | Ответ на вопрос не является полным. Материал изложен непоследовательно. Выводы не аргументированы. На один из двух дополнительных вопросов не дано удовлетворительного ответа. |
| «Неудовлетворительно» (0-3) | Ответ на вопрос является неверным. Материал изложен непоследовательно. Отсутствуют выводы. Ни на один из дополнительных вопросов не дано удовлетворительного ответа. |

## 7.3 Порядок формирования оценок по дисциплине

Преподаватель учитывает оценку за текущий контроль (домашние задания).

О*накопленная* =

Действует следующий способ округления накопленной оценки текущего контроля: при значениях от 0,1 до 0,4 оценка округляется в меньшую сторону, от 0,5 до 0,9 – в большую.

Результирующая оценка за дисциплину рассчитывается следующим образом:

О*Результирующая=* 0,5 О*накопленная+0,5* О*экзамен*

На экзамене студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

# Образовательные технологии

# Основными образовательными технологиями являются: интерактивные лекции, работа в группах на практических занятиях.

# Учебно-методическое и информационное обеспечение дисциплины

## 9.1 Основная литература

1. Joseph Albahari, Ben Albahari, C# 7.0 in a Nutshell, O'Reilly Media, 2017, 1088pp.
2. J. Richter, CLR via C# (4th edition), Microsoft Press, 2012, 896pp.
3. Stephen Cleary, Concurrency in C# Cookbook, O’Reilly Media, 2014. 207pp.
4. Don Syme, Adam Granicz, Antonio Cisternino, Expert F# 4.0, Apress, 2015, 582pp.

## Дополнительная литература

1. Репозиторий документации по .NET, URL: <https://github.com/dotnet/docs/>
2. Блог “FSharp for fun and profit”, URL: <https://fsharpforfunandprofit.com/>
3. Adam Nathan, WPF 4.5 Unleashed. Sams Publishing, 2013. 864pp.

# Рекомендации для самостоятельной работы студентов

Самостоятельная работа может рассматриваться как организационная форма обучения – система педагогических условий, обеспечивающих управление учебной деятельностью по освоению знаний и умений в области учебной деятельности без посторонней помощи. Студенту нужно четко понимать, что самостоятельная работа – не просто обязательное, а необходимое условие для получения знаний по дисциплине и развитию компетенций, необходимых в будущей профессиональной деятельности.

Самостоятельная работа проводится с целью:

* систематизации и закрепления полученных на лекциях теоретических знаний;
* углубления и расширения теоретических знаний;
* формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
* развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
* формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
* формирования практических (общеучебных и профессиональных) умений и навыков;
* развития исследовательских умений;
* получения навыков эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности.

В учебном процессе выделяют два вида самостоятельной работы:

* аудиторная;
* внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа - планируемая учебная работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная программой учебной дисциплины, раскрывающей и конкретизирующей ее содержание, осуществляется студентом инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует источники для работы, характеризует наиболее рациональную методику самостоятельной работы, демонстрирует ранее выполненные студентами работы и т. п.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать индивидуальные особенности студента.

Самостоятельная работа может осуществляться индивидуально или группами студентов online и на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности.

Контроль результатов внеаудиторной самостоятельной работы осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине на практических занятиях.

# Материально-техническое обеспечение дисциплины и информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения информационных справочных систем (при необходимости).

Для проведения всех занятий используется проектор и компьютер для проекции слайдов. Для самостоятельной работы необходимо следующее ПО: .NET Framework версии не ниже 4.5 либо .NET Core версии не ниже 2.0.

# Особенности организации обучения для лиц с ограниченными возможностями здоровья

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться следующих варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

1) *для лиц с нарушениями зрения:* в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

2) *для лиц с нарушениями слуха*: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

3) *для лиц с нарушениями опорно-двигательного аппарата*: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.