**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Введение в MS.NET

Introduction to Microsoft.NET

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 2

Регистрационный номер рабочей программы: 003739

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Цель изучения дисциплины: ознакомить обучающихся с современной технологией разработки программного обеспечения Microsoft.NET. Весь материал снабжается задачами, которые обучающиеся в качестве обязательных домашних заданий выполняют в течение курса.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа курса предназначена для обучающихся 4 курса. Курс рассчитан на обучающихся, имеющих начальные навыки объектно-ориентированного программирования и реализовавших в своей жизни самостоятельно хотя бы одно законченное программное приложение. Предполагается, что данные навыки обучающиеся получили на практикуме по программированию на первых курсах.

**1.3. Перечень результатов обучения (learning outcomes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование категории (группы) компетенций | Код и наименование компетенции | Планируемые результаты обучения, обеспечивающие формирование компетенции | Код индикатора и индикатор достижения универсальной компетенции |
|  | 1 | 2 | 3 | 4 |
| 1 | Общепрофессиональные компетенции | ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности | Базовое понимание принципов платформы .NET и особенностей работы с ней | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения |
| 2 | Общепрофессиональные компетенции | ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности | Базовое понимание принципов и алгоритмов работы сборщика мусора. | ОПК-2.1 Уметь писать программный код с использованием языков программирования, определения и манипулирования данными |
| 3 | Профессиональные компетенции | ПКП-1 Способен проектировать программные системы | Навыки написания кода с использованием языка C# и платформы .NET, работы с многопоточным кодом, с продуктом Microsoft Visual Studio .NET Community Edition. | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие |
| 4 | Профессиональные компетенции | ПКП-2 Способен использовать основные модели информационных технологий и способы их применения для решения задач в предметных областях | Навыки написания кода с использованием языка C# и платформы .NET, работы с многопоточным кодом, с продуктом Microsoft Visual Studio .NET Community Edition. | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы |
| 5 | Профессиональные компетенции | ПКП-3 Способен разрабатывать моделирующие алгоритмы и реализации их на базе языков и пакетов прикладных программ моделирования | Навыки написания кода с использованием языка C# и платформы .NET, работы с многопоточным кодом, с продуктом Microsoft Visual Studio .NET Community Edition. | ПКП-3.1 Разработка модели бизнес-процессов заказчика |
| 6 | Профессиональные компетенции | ПКП-7 Способен систематизировать и применять знания о содержании основных этапов и тенденций развития программирования, математического обеспечения и информационных технологий | Навыки написания кода с использованием языка C# и платформы .NET, работы с многопоточным кодом, с продуктом Microsoft Visual Studio .NET Community Edition. | ПКП-7.1 Умеет проводить технические и управленческие ревизии создаваемого программного средства |
| 7 | Универсальные компетенции | УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач | Навыки активного общения – как с друг другом, так и с преподавателем. | УК 1.2. Определяет информацию, необходимую для решения поставленной задачи; |
| 8 | Универсальные компетенции | УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений | Навыки активного общения – как с друг другом, так и с преподавателем. | УК-2.5. Выполняет задачи в зоне своей ответственности в соответствии с запланированными результатами и точками контроля; |
| 9 | Универсальные компетенции | УКБ-1 Способен участвовать в разработке и реализации проектов, в т.ч. предпринимательских | Навыки активного общения – как с друг другом, так и с преподавателем. | УКБ-1.3. Оценивает соответствие способов решения цели проекта; |
| 10 | Универсальные компетенции | УКБ-2 Способен устанавливать и поддерживать взаимоотношения в социальной и профессиональной сфере, исходя из нетерпимости к коррупционному поведению и проявлениям экстремизма | Навыки активного общения – как с друг другом, так и с преподавателем. | УКБ-3.4. Проверяет достоверность собранной информации. |

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Семинары – 20 ак. ч.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 7 |  | 30 |  |  |  |  |  |  | 2 |  |  |  | 33 |  | 7 |  | 20 | 2 |
|  |  | 2-25 |  |  |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО |  | 30 |  |  |  |  |  |  | 2 |  |  |  | 33 |  | 7 |  |  | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 7 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

**Основной курс Основая траектория Очная форма обучения**

Период одбучения (модуль): Семестр 7

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| 1. | Введение в .NET | Семинары | 2 |
| 2. | Обзор языка C# | Семинары | 4 |
| 3. | Система типов в .NET. Массивы, строки. | Семинары | 4 |
| 4. | Эволюция платформы .NET и языка C# | Семинары | 2 |
| 5. | Сборки. Развёртывание приложений в .NET. Домены приложений | Семинары | 2 |
| Самостоятельная работа с методическими материалами | 5 |
| 6. | Многопоточность | Семинары | 8 |
| Самостоятельная работа с методическими материалами | 10 |
| 7. | Сборщик мусора | Семинары | 4 |
| Самостоятельная работа с методическими материалами | 6 |
| 8. | Стек технологий .NET | Семинары | 4 |
| Самостоятельная работа с методическими материалами | 12 |
| 9. | Промежуточная аттестация | Промежуточная аттестация (зачёт) | 2 |
| Промежуточная аттестация (самостоятельная работа) | 7 |

На семинарах преподаватель рассказывает материал курса согласно следующему содержанию в разбивке по разделам:

1. **Введение в .NET**

История появления Microsoft .NET. Предпосылки. Основные задачи, стоявшие перед разработчиками .NET. Архитектура .NET. Обзор Common Language Runtime. Особенности Microsoft Intermediate Language. Его роль в работе .NET-приложений. Обзор Framework Class Library.

1. **Обзор языка C#**

Определение языка C# как современного объектно-ориентированного языка программирования со строгой статической типизацией. Операторы в языке C#.Классы и структуры. Пространства имён. Модификаторы доступа. Ограничения по сравнению с CLR. Статические и динамические константы: const и readonly. Особенности. Наследование. Переопределение операторов: явное и неявное. Свойства и индексаторы. Передача неизвестного числа аргументов в метод: ключевое слово params.

1. **Система типов в .NET. Массивы, строки**

System.Object как базовый тип для всех типов в .NET. Обзор его методов и особенности их реализации. Краткий обзор иерархии типов .NET (Basic Class Library). Типы-значения. Особенности реализации. Ссылочные типы. Особенности реализации. Передача параметра по ссылке. Упаковка и распаковка типа-значений. Особенности реализации и типичные проблемы. Виды массивов в .NET: массивы с нулевой и произвольной нижней границей, рваные массивы, многомерные массивы. Особенности реализации. Работа с массивами с использованием неуправляемого кода (unsafe). Специальные методы для работы с массивами: копирование с использованием проверки типов, блочное копирование, копирование как транзакция. Строки в .NET. Внутреннее устройство. Особенности работы со строками. Механизм интернирования. Дополнительные классы для строк: StringBuilder и SecureString. Особенности работы с ними.

1. **Эволюция платформы .NET и языка C#**

Основные нововведения .NET 2.0 и C# 2.0: обобщения, анонимные методы, частичные классы, nullable типы. Пул потоков. Изменения в обработке исключений. Основные нововведения .NET 3.0 и 3.5, а также C# 3.0: автоматический вывод типов (var); инициализаторы объектов и коллекций; анонимные типы; лямбда-выражения, LINQ; деревья выражений. Основные нововведения .NET 4.0 и 4.5, а также C# 4.0 и 5.0: Dynamic Language Runtime и ключевое слово dynamic; ковариантность и контрвариантность обобщений; опциональные параметры в C#; значение по умолчанию для пареаметров метода; инструментарий для параллельных вычислений – PLINQ, Task Parallel Library (TPL) и дальнейшее её развитие в TPL Dataflow; асинхронные вычисления (async и await); улучшения в работе сборщика мусора. Будущее .NET и C#.

1. **Сборки. Развёртывание приложений в .NET. Домены приложений**

Понятие сборки как атомарной единицы приложения в .NET. Характеристики сборки. Проблема DLL Hell. Сборки со строгим именем. Глобальный кэш сборок. Развёртывание приложений в .NET. Управление зависимостями. Мониторинг .NET приложений. Понятие домена приложения. Основные концепции и цели реализации. Особенности работы с доменами приложений в .NET.

1. **Многопоточность**

Краткое введение в многопочность. Процессы и потоки в операционной системе. Задачи, решаемые с помощью многопоточности. Основные проблемы многопоточности. Класс Thread в .NET и особенности работы с ним. Пул потоков и особенности работы с ним. Изменение парадигмы работы с многопоточным кодом – от потоков к задачам: использование Task Parallel Library. Класс Task. Инструментарий PLINQ и особенности его реализации. Методы распределения задач по потокам, механизмы «слияния» последовательных задач, способы управления ходом исполнения запроса. Дальнейшее развитие идеи распараллеливания задач - обзор TPL Dataflow. Использование async и await для выполнения асинхронных операций. Синхронизация в многопоточных прилоениях. Механизмы синхронизации и их особенности: мьютексы, события, семафоры, мониторы, атомарные (interlocked) функции. Ключевое слово lock и особенности его реализации в различных версиях C#. Особенности реализации синхронизации в .NET. Использование коллекций в многопоточных приложениях. Семейство Concurrent коллекций.

1. **Сборщик мусора**

Управляемые и неуправляемые приложения. Динамическая память и стек. Модель памяти в .NET. Специальные участки динамической памяти в .NET. Внутреннее устройство объектов в .NET. Сборщик мусора: основные задачи и известные проблемы. Алгоритм работы сборщика мусора. Очистка неуправляемых ресурсов. Финализаторы и их влияние на сборку мусора. IDisposable. Улучшения сборщика мусора в разлчных версиях .NET.

1. **Стек технологий .NET.**

WinForms. Форма. Управляющие элементы. Привязка данных. WPF. XAML-разметка. Паттерн MVVM. Привязка. Понятие шаблона в WPF. Понятие триггера в WPF. WCF. Адрес, контракт, привязка.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Для освоения дисциплины обучающиеся должны посещать семинары и выполнять задания преподавателя для самостоятельной работы.

Материал семинаров сопровождается практическими заданиями для модулей, сопровождающихся самостоятельной работой. Всем обучающимся выдаются одни и те же задания, но выполнять их необходимо строго индивидуально. Рекомендуется давать по одной задаче на каждый такой модуль. Пример задания по модулю «Многопоточность»:

Реализовать объект ThreadPool, реализующий паттерн «пул потоков». Число потоков задаётся константой в классе пула. Добавление задачи осуществляется с помощью нестатического метода класса пула Enqueue(Action a). Класс должен быть унаследован от интерфейса IDisposable и корректно освобождать ресурсы при вызове метода Dispose().

**3.1.2 Методическое обеспечение самостоятельной работы**

При самостоятельном изучении теоретического материала, рассказываемого на семинарах и выполнении практических заданий целесообразно использовать рекомендованную основную и дополнительную литературу. По согласованию с преподавателем обучающийся может осваивать теоретическую часть курса по одному из онлайн-курсов, представленных на образовательных платформах Coursera, EDX или Stepik. При этом преподаватель обязан сообщить обучащимся, какие разделы выбранного им онлайн-курса недостаточно полно раскрывают ту или иную тему курса и порекомендовать дополнительные источники по данной теме. В случае, если преподавателя не устраивает результат самостоятельного освоения дисциплины теоретической части дисциплины по онлайн-курсу в случае нахождения в нём существенных расхождений с содержанием курса в разделе 2 и п. 3.1.1, преподаватель вправе не зачесть этот результат. В таком случае преподаватель в индивидуальном порядке обсуждает с обучающимся дополнительные шаги по освоению дисциплины.

Задания преподавателя по отдельным модулям дисциплины могут быть полностью выполнена самостоятельно по методическим материалам.

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Промежуточная аттестация состоит из прохождения текущего контроля успеваемости и устного зачёта.

Текущий контроль успеваемости состоит в выполнении всех заданий для самостоятельной работы, предлагаемых обучающимся в течение семестра. Успешная сдача каждой задачи оценивается в 10 баллов. По желанию преподавателя можно добавлять дополнительные условия:

* + Некоторые задачи могут быть повышенной сложности и оцениваться в 20 баллов.
  + Для задач допустимо вводить промежуточные и предельные сроки сдачи. Сдача после промежуточного срока оценивается не более чем в половину максимальной оценки за задачу, после предельного срока – строго в 0 баллов.
  + В начале семестра преподаватель сообщает обучающимся правила хорошего оформления программ, соответствующие ведущим практикам индустрии разработки программного обеспечения. Обучающийся теряет 3 балла за каждую попытку сдачи с нарушениями этих правил (вплоть до отрицательной оценки за задачу).
  + Баллы за задачу могут быть снижены вплоть до 50% (с округлением вниз), если тестовое покрытие решения недостаточно.
  + Баллы за задачу могут быть снижены до 50% (с округлением вниз), если решение существенно неоптимально.

Билет на устном зачёте состоит из одного вопроса. Преподаватель вправе выбрать одну из форм проведения зачёта:

* классическая форма, при которой время подготовки ответа на вопрос билета составляет не менее 0,5 академического часа, и на время ответа ограничения не накладывается;
* блиц-опрос, когда обучающийся начинает сразу же после получения билета, а время ответа ограничивается 10-15 минутами.

В обоих случаях использование конспектов и учебников, а также электронных устройств хранения, обработки или передачи информации при подготовке и ответе на вопросы зачета запрещено.

После ответа на вопрос билета, преподаватель вправе задать уточняющие вопросы по услышанному. Затем преподаватель задает дополнительные вопросы по любой теме из списка вопросов, вынесенных на зачет. В качестве уточняющих используются вопросы, не требующие длительного ответа, в том числе основные определения и понятия. Дополнительные вопросы должны строиться таким образом, чтобы с их помощью обучающийся мог продемонстрировать своё понимание или непонимание материала, а также способности к аргументированному обсуждению. Рекомендуется задавать 3-4 дополнительных вопроса на билет и не более 2-3 уточняющих вопросов. Преподаватель вправе увеличить количество вопросов в случае, если у него не возникает понимание, освоил обучающийся материал учебного курса или нет.

В любой момент обучающийся имеет право отказаться от ответа с выставлением оценки в 0 баллов.

Максимальная оценка K за устный зачёт равна сумме максимальных баллов за задания для самостоятельной работы в течение семестра плюс один балл. Максимальный балл за основной и уточняющие вопросы не может составлять более 50% K. Оставшийся бюджет из 50% баллов делится поровну по дополнительным вопросам. Преподаватель вправе снижать баллы за неточности и ошибки в зависимости от их грубости. Если суммарная оценка за устный зачёт превышает порог в 70% К, считается, что обучающийся сдал устный зачёт на полученное количество баллов, которые добавляются к баллам, накопленным в семестре. В противном случае считается, что обучающийся сдал зачёт на 0 баллов.

По желанию преподавателя на экзамен допустимо приглашать других преподавателей с квалификацией не ниже изложенной в п. 3.2.1 как для независимого оценивания ответов обучающихся, так и для коллегиального. В последнем случае оценка за зачет ставится на основании голосования простого большинства. В спорных ситуациях преподаватель, ведущий дисциплину, имеет право принятия окончательного решения.

Перевод из баллов, полученных за мероприятия текущего и промежуточного контроля, в итоговый процент освоения курса делается по формуле

где I – итоговый процент освоения курса, N – максимальное количество баллов, зарабатываемое в течение семестра, n – заработанное обучающимся число баллов. Перевод в оценки делается по стандартной методике согласно приказу №7293/1 от 20.07.2018:

|  |  |  |
| --- | --- | --- |
| Итоговый процент освоения курса, % | Оценка СПбГУ при проведении зачёта | Оценка ECTS |
| 0-49 | Не зачтено | F |
| 50-60 | Зачтено | E |
| 61-69 | Зачтено | D |
| 70-79 | Зачтено | C |
| 80-89 | Зачтено | B |
| 90-100 | Зачтено | A |

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

Список вопросов на зачёт:

* + 1. Архитектура .NET. Common Language Runtime. Common Intermediate Language. Framework Class Library.
    2. Классы и структуры. Модификаторы доступа. Ограничения по сравнению с CLR. Свойства.
    3. Операторы в языке C#. Пространства имён. Статические и динамические константы: const и readonly. Особенности.
    4. System.Object как базовый тип для всех типов в .NET. Краткий обзор иерархии типов .NET (Basic Class Library).
    5. Типы-значения. Особенности реализации. Упаковка и распаковка типа-значений. Особенности реализации и типичные проблемы.
  1. Ссылочные типы. Особенности реализации. Передача параметра по ссылке.

1. Наследование в C#.
2. Обработка исключений в .NET.
3. Виды массивов в .NET: массивы с нулевой и произвольной нижней границей, рваные (jagged) массивы, многомерные массивы. Особенности реализации.
4. Обобщенное программирование в C#.
5. ковариантность и контрвариантность обобщений;
6. Делегаты и события в .NET.
7. LINQ.
8. Модель памяти в .NET. Динамическая память и стек. Внутреннее устройство объектов в .NET.
9. Сборщик мусора: основноые задачи и известные проблемы. Алгоритм работы сборщика мусора. Улучшения сборщика мусора.
10. Очистка неуправляемых ресурсов. Финализаторы и их влияние на сборку мусора. IDisposable.
11. Понятие домена приложения. Основные концепции и цели реализации. Особенности работы с доменами приложений в .NET.
12. Понятие маршалинга. Сценарии использования.
13. Краткое введение в многопочность. Процессы и потоки в операционной системе. Задачи, решаемые с помощью многопоточности. Основные проблемы многопоточности. Класс Thread в .NET и особенности работы с ним. Пул потоков и особенности работы с ним.
14. Task Parallel Library. Класс Task. Использование async и await для выполнения асинхронных операций.
15. Синхронизация в многопоточных приложениях. Механизмы синхронизации и их особенности: мьютексы, события, семафоры, мониторы, атомарные (interlocked) функции. Ключевое слово lock и особенности его реализации.
16. Основные понятия WPF. XAML-разметка. Паттерн MVVM. Привязка.
17. Понятие шаблона в WPF. Понятие триггера в WPF.
18. Основные понятия WCF. Адрес, контракт, привязка.

|  |  |  |
| --- | --- | --- |
| № | Код индикатора и индикатор достижения универсальной компетенции | Контрольно-измерительные материалы (КИМ) (тестовые вопросы, контрольные задания, кейсы и пр.) |
|  | 1 | 2 |
| 1 | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения | итоговый процент освоения курса |
| 2 | ОПК-2.1 Уметь писать программный код с использованием языков программирования, определения и манипулирования данными | процент набранных баллов за практические задачи от максимального |
| 3 | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие | процент набранных баллов за практические задачи от максимального |
| 4 | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы | процент набранных баллов за теоретический зачёт от максимального |
| 5 | ПКП-3.1 Разработка модели бизнес-процессов заказчика | процент набранных баллов за практические задачи от максимального |
| 6 | ПКП-7.1 Умеет проводить технические и управленческие ревизии создаваемого программного средства | итоговый процент освоения курса |
| 7 | УК 1.2. Определяет информацию, необходимую для решения поставленной задачи; | процент набранных баллов за теоретический зачёт от максимального |
| 8 | УК-2.5. Выполняет задачи в зоне своей ответственности в соответствии с запланированными результатами и точками контроля; | процент набранных баллов за практические задачи от максимального |
| 9 | УКБ-1.3. Оценивает соответствие способов решения цели проекта; | итоговый процент освоения курса |
| 10 | УКБ-2.4. Проверяет достоверность собранной информации. | итоговый процент освоения курса |

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем техническом образовании по направлениям «Программная инженерия», «Математическое обеспечение и администрирование информационных систем» и другим смежным направлениям, связанным с информационными технологиями.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

При проведении занятий в компьютерных классах (см. п. 3.3.1) для подготовки и проведения занятий требуется лаборант или инженер для следующих работ:

* + 1. Техническая подготовка каталогов исходных данных в форме, удобной для учебной работы.
    2. Настройка пакетов прикладных программ, перечисленных в п. 3.3.2
    3. Поддержание работоспособности компьютерного класса

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

Учебные аудитории для проведения учебных занятий, оснащенные стандартным оборудованием, используемым для обучения в СПбГУ в соответствии с требованиями материально-технического обеспечения.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Стандартное оборудование, используемое для обучения в СПбГУ. MS Windows, MS Office, Mozilla FireFox, Google Chrome, Acrobat Reader DC, WinZip, Антивирус Касперского.

**3.3.3 Характеристики специализированного оборудования**

Специализированное оборудование не требуется.

**3.3.4 Характеристики специализированного программного обеспечения**

При проведении в компьютерном классе на каждом компьютере для обучающихся должны быть установлены следующие программные системы:

MS Visual Studio 2013 Community Edition или выше

JetBrains DotPeek 1.2

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Для аудиторий с маркерными досками необходимы стирающиеся маркеры в объеме, достаточном для проведения курса. Для аудиторий с меловыми досками необходим мел в объеме, достаточном для проведения курса.

**3.4. Информационное обеспечение**

**3.4.1 Список литературы**

1. Терехов, Андрей Николаевич. Технология программирования: учебное пособие / А. Н. Терехов. - 2-е изд. - М.: Интернет-Университет Информационных Технологий; М. : БИНОМ. Лаборатория знаний, 2007-2017. - 149 с.

2. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. — 4-е изд. — СПб: Питер, 2013-2018. — 896 с.

[**https://proxy.library.spbu.ru:2374/bookshelf/338551/reading**](https://proxy.library.spbu.ru:2374/bookshelf/338551/reading)

**3.4.2 Перечень иных информационных источников**

• <http://ericlippert.com/>

• Сайт Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/>

• Электронный каталог Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/cgi-bin/irbis64r/cgiirbis_64.exe?C21COM=F&I21DBN=IBIS&P21DBN=IBIS>

• Перечень электронных ресурсов, находящихся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/>

• Перечень ЭБС, на платформах которых представлены российские учебники, находящиеся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/browse?name=rures&resource%20type=8>

**Раздел 4. Разработчики программы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ФИО | Ученая  степень | Ученое звание | Должность | Контакты |
| Сартасов  Станислав Юрьевич |  |  | ст.преподаватель кафедры системного программирования | s.sartasov@spbu.ru |