

МИНОБРНАУКИ РОССИИ



Федеральное государственное бюджетное образовательное учреждение
высшего образования
«**Российский государственный гуманитарный университет**»
(ФГБОУ ВО «РГГУ»)

ИНСТИТУТ ИНФОРМАЦИОННЫХ НАУК И ТЕХНОЛОГИЙ БЕЗОПАСНОСТИ
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И БЕЗОПАСНОСТИ
Кафедра информационных технологий и систем

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

09.03.03 Прикладная информатика

Код и наименование направления подготовки/специальности

Прикладная информатика в гуманитарной сфере

Наименование направленности (профиля)/ специализации

Уровень высшего образования: *бакалавриат*

Форма обучения: *очная*

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2023

Объектно-ориентированное программирование
Рабочая программа дисциплины

Составитель:

старший преподаватель кафедры информационных технологий и систем Е.П. Охапкина

Ответственный редактор

*канд. с.-х. наук, доцент, заведующий кафедрой
информационных технологий и систем Н.Ш. Шукенбаева*

УТВЕРЖДЕНО

Протокол заседания кафедры ИТС

№ 8 от 15.04.2023 г.

ОГЛАВЛЕНИЕ

1.	Пояснительная записка	4
1.1.	Цель и задачи дисциплины	4
1.2.	Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций	4
1.3.	Место дисциплины в структуре образовательной программы	5
2.	Структура дисциплины	5
3.	Содержание дисциплины	6
4.	Образовательные технологии	7
5.	Оценка планируемых результатов обучения	7
5.1	Система оценивания	7
5.2	Критерии выставления оценки по дисциплине	8
5.3	Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине	9
6.	Учебно-методическое и информационное обеспечение дисциплины	10
6.1	Список источников и литературы	10
6.2	Перечень ресурсов информационно-телекоммуникационной сети «Интернет».	11
6.3	Профессиональные базы данных и информационно-справочные системы	11
7.	Материально-техническое обеспечение дисциплины	11
8.	Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов	12
9.	Методические материалы	13
9.1	Планы практических занятий	13
9.2	Методические рекомендации по подготовке письменных работ	14
	Оформление отчета выполняется с использованием компьютерной верстки LaTeX. Отчет сохраняется и представляет для проверки в виде отдельного pdf файла. В имени файла указывается фамилия студента и номер выполненной работы.	15
9.3	Рекомендации по написанию курсовой работы	15
	Приложение 1. Аннотация дисциплины	17

1. Пояснительная записка

1.1. Цель и задачи дисциплины

Цель дисциплины – приобретение знаний, навыков и умений в области программирования на языке высокого уровня C#, а также освоение основных алгоритмов обработки и хранения больших данных.

Задачи дисциплины:

1. Изучение базовых принципов программирования на языке высокого уровня C#;
2. Дать широко применяемые технологии и методы программирования для разработки приложений на языке C#, в том числе, специализированные методы обработки и хранения данных;
3. Изучение главных управляющих структур языков C#;
4. Изучение принципов объектно-ориентированного программирования. Понятие о классе и его свойствах;
5. Сформировать навыки и умения по разработке алгоритмов в задачах последовательного и объектно-ориентированного программирования;
6. Познакомить со стандартами языков C# и их отличительными особенностями;
7. Дать представления о процедуре отладки разрабатываемых приложений;
8. Сформировать представления о разработке эффективных приложений и обучить основным методам оптимизации.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
<i>ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения</i>	<i>ОПК-7.1 Знает основные языки программирования, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий</i>	<i>Знать: базовые принципы программирования на языке высокого уровня; главные управляющие структуры языка C#; принципы объектно-ориентированного программирования; алгоритмы отладки разрабатываемых приложений; методы разработки эффективных приложений и оптимизации по времени и памяти; основы построения самообучающихся алгоритмов; специализированные методы обработки и хранения данных.</i>
	<i>ОПК-7.2 Умеет применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач</i>	<i>Уметь: применять специализированные методы обработки и хранения данных на языке C#; разрабатывать алгоритмы для решения задач последовательного и объектно-ориентированного программирования; применять принципы объектно-ориентированного программирования для построения высоко эффективных приложений; разрабатывать простейшие модели реляционных баз данных на языке высокого уровня C#;</i>

	<i>различных классов, ведения баз данных и информационных хранилищ</i>	
	<i>ОПК-7.3 Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач</i>	<i>Владеть: базовыми принципами программирования на языке высокого уровня; методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение программы и используемой памяти компьютера; основными управляющими структурами языка С#; методами сортировки и хранения данных; методами и подходами к динамическому выделению памяти в пространстве памяти компьютера.</i>

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование С#» относится к обязательной части блока дисциплин учебного плана.

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения школьного курса Информатика.

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: Программная инженерия, Программирование пользовательских интерфейсов информационных систем гуманитарной сферы.

2. Структура дисциплины

Общая трудоёмкость дисциплины составляет:

- в первом семестре 3 з.е., 72 академических часа.
- во втором семестре 4з.е., 216 академических часа.

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
1	Лекции	12
1	Практические занятия	16
Всего:		28

Семестр	Тип учебных занятий	Количество часов
2	Лекции	16
2	Практические занятия	30
Всего:		46

Объем дисциплины (модуля) в форме самостоятельной работы обучающихся составляет 44 академических часа в первом семестре и 168 академических часа во втором семестре.

3. Содержание дисциплины

№	Наименование раздела дисциплины	Содержание
Семестр 1		
1	Основы программирования на языке C#.	Отличительные особенности C#. Объектно-ориентированное программирование. Примитивные типы. Числа с плавающей точкой. Символы. Логические значения. Переменные. Преобразование и приведение типов. Автоматическое продвижение типов в выражениях. Массивы. Арифметические операции. Поразрядные операции. Операции отношения. Операция присваивания. Тернарная операция ?. Операторы выбора. Условный оператор if . Оператор switch . Операторы цикла Цикл while . Цикл do...while . Цикл for . Вложенные циклы. Операторы перехода. Применение оператора break . Применение оператора continue . Оператор return .
2	Классы. Наследование. Многопоточное программирование C#.	Основы классов. Общая форма класса. Простой класс. Объявление объектов. Подробное рассмотрение оператора new . Присваивание переменным ссылок на объекты. Введение в методы. Возврат значений. Ввод метода, принимающего параметры. Конструкторы. Параметризованные конструкторы. Ключевое слово this . Соккрытие переменной экземпляра. Сборка "мусора". Перегрузка методов. Перегрузка конструкторов. Применение объектов в качестве параметров. Подробное рассмотрение особенностей передачи аргументов. Возврат объектов. Рекурсия. Введение в управление доступом. Ключевое слово static . Ключевое слово final . Основы наследования. Доступ к членам класса и наследование. Создание многоуровневой иерархии. Порядок вызова конструкторов. Переопределение методов. Применение абстрактных классов. Модель потоков исполнения в C#. Главный поток исполнения. Создание потока исполнения. Создание многих потоков исполнения.
3	Лямбда-функции. Введение в библиотеку C#.	Введение в лямбда-выражения. Блочные лямбда-выражения. Обобщенные функциональные интерфейсы. Передача лямбда-выражений в качестве аргументов. Лямбда-выражения и исключения. Лямбда-выражения и захват переменных. Ссылки на методы. Ссылки на конструкторы. Конструкторы символьных строк. Длина

		символьной строки. Специальные строковые операции. Извлечение символов. Сравнение символьных строк. Поиск в символьных строках.
4	Чтение и запись данных в файлы. Автоматическое закрытие файла.	Основы ввода-вывода. Потоки ввода-вывода. Потоки ввода-вывода байтов и символов. Предопределенные потоки ввода-вывода. Чтение данных, вводимых с консоли. Запись данных, выводимых на консоль. Чтение и запись данных в файлы. Автоматическое закрытие файла.
5	Чтение и запись данных в файлы. Автоматическое закрытие файла. Основы создания апплетов.	Основы создания апплетов. Модификаторы доступа. Платформенно-ориентированные методы. Трудности, связанные с платформенно-ориентированными методами. Применение ключевого слова assert . Параметры включения и отключения режима проверки утверждений. Статический импорт. Вызов перегружаемых конструкторов по ссылке this() . Компактные профили C# API .
Семестр 2		
6.	Использование стандартных интерфейсов	реализация, получение ссылки, интерфейс как параметр, явная реализация, создание иерархий, наследование от нескольких базовых интерфейсов, интерфейсные индексы
7.	Делегаты	использование, делегаты как статические члены класса
8.	Перегрузка операторов	операторы отношения, приведения, примеры, унарные, бинарные операторы, пример использования перегрузки операторов, ограничения
9.	Компоненты	определение, IComponent, класс Component, простой компонент, использование, переопределение метода Dispose()

4. Образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.– видео-лекции;

5. Оценка планируемых результатов обучения

5.1 Система оценивания

Семестр 1:

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
- Практическая работа №1	12 баллов	12 баллов
- Практическая работа №2	12 баллов	12 баллов

- Практическая работа №3	12 баллов	12 баллов
- Практическая работа №4	12 баллов	12 баллов
- Практическая работа №5	12 баллов	12 баллов
Промежуточная аттестация: зачет		40 баллов
Итого за семестр экзамен		100 баллов

Семестр 2:

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
Практическая работа № 6, защита отчета	15 баллов	15 баллов
Практическая работа № 7, защита отчета	15 баллов	15 баллов
Практическая работа № 8, защита отчета	15 баллов	15 баллов
Практическая работа № 9, защита отчета	15 баллов	15 баллов
Промежуточная аттестация экзамен		40 баллов
Итого за семестр		100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	отлично	зачтено	A
83 – 94			B
68 – 82	хорошо		C
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	не зачтено	FX
0 – 19			F

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A, B	отлично/ зачтено	<p>Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения.</p> <p>Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
82-68/ C	хорошо/ зачтено	Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей. Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами. Достаточно хорошо ориентируется в учебной и профессиональной литературе. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».
67-50/ D,E	удовлетво- рительно/ зачтено	Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами. Демонстрирует достаточный уровень знания учебной литературы по дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».
49-0/ F,FX	неудовлет- ворительно/ не зачтено	Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами. Демонстрирует фрагментарные знания учебной литературы по дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.

5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Вопросы для подготовки к экзамену (ОПК-7)

1. Отличительные особенности C#.
2. Объектно-ориентированное программирование.
3. Прimitивные типы. Числа с плавающей точкой. Символы.
4. Логические значения. Переменные. Преобразование и приведение типов. Автоматическое продвижение типов в выражениях.
5. Массивы. Арифметические операции. Поразрядные операции. Операции отношения.
6. Операция присваивания. Тернарная операция ?. Операторы выбора.
7. Условный оператор if. Оператор switch.
8. Операторы цикла. Цикл while. Цикл do...while. Цикл for.
9. Вложенные циклы. Операторы перехода. Применение оператора break.
10. Применение оператора continue. Оператор return.
11. Основы классов. Общая форма класса. Простой класс. Объявление объектов.
12. Подробное рассмотрение оператора new. Присваивание переменным ссылок на объекты.
13. Введение в методы. Возврат значений. Ввод метода, принимающего параметры.

14. Конструкторы. Параметризованные конструкторы.
15. Ключевое слово `this`. Скрытие переменной экземпляра. Сборка "мусора".
16. Перегрузка методов. Перегрузка конструкторов.
17. Применение объектов в качестве параметров.
18. Возврат объектов. Рекурсия. Введение в управление доступом.
19. Ключевое слово `static`. Ключевое слово `final`.
20. Основы наследования. Доступ к членам класса и наследование.
21. Создание многоуровневой иерархии.
22. Порядок вызова конструкторов. Переопределение методов.
23. Применение абстрактных классов.
24. Модель потоков исполнения в C#. Главный поток исполнения.
25. Создание потока исполнения. Создание многих потоков исполнения.
26. Введение в лямбда-выражения. Блочные лямбда-выражения.
27. Обобщенные функциональные интерфейсы. Передача лямбда-выражений в качестве аргументов.
28. Лямбда-выражения и исключения. Лямбда-выражения и захват переменных.
29. Ссылки на методы. Ссылки на конструкторы.
30. Конструкторы символьных строк. Длина символьной строки.
31. Специальные строковые операции. Извлечение символов.
32. Сравнение символьных строк. Поиск в символьных строках.
33. Основы ввода-вывода. Потоки ввода-вывода.
34. Потоки ввода-вывода байтов и символов. Предопределенные потоки ввода-вывода.
35. Чтение данных, вводимых с консоли. Запись данных, выводимых на консоль.
36. Чтение и запись данных в файлы.
37. Автоматическое закрытие файла.
38. Основы создания апплетов. Модификаторы доступа.
39. Платформенно-ориентированные методы.
40. Трудности, связанные с платформенно-ориентированными методами.
41. Параметры включения и отключения режима проверки утверждений.
42. Статический импорт. Вызов перегружаемых конструкторов по ссылке `this()`.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Список источников и литературы

Литература

Основная

1. Хорев, П. Б. Объектно-ориентированное программирование с примерами на C# : учебное пособие / П.Б. Хорев. — Москва : ФОРУМ : ИНФРА-М, 2023. — 200 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-680-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1926392>
2. Подбельский, В. В. Программирование. Базовый курс C#: учебник для вузов / В. В. Подбельский. — Москва: Издательство Юрайт, 2020. — 369 с. — URL: <https://urait.ru/bcode/450868>.
3. Казанский, А. А. Программирование на Visual C#: учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2020. — 192 с. — URL: <https://urait.ru/bcode/451467>.
4. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C#: учебное пособие для вузов / Е. В. Кудрина, М. В. Огнева. — Москва: Издательство Юрайт, 2020. — 322 с. - URL: <https://urait.ru/bcode/456182>.

Дополнительная

1. Тузовский, А. Ф. Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — Москва: Издательство Юрайт, 2018. — URL: <https://urait.ru/bcode/414163>.
2. Гуриков, С. Р. Введение в программирование на языке Visual C#: учеб. пособие / С.Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2017. — 447 с. - URL: <https://znanium.com/catalog/product/752394>.
3. Гниденко, И. Г. Технологии и методы программирования: учебное пособие для прикладного бакалавриата / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2017. — 235 с. — URL: <https://urait.ru/bcode/401989>.
4. Подбельский, В. В. Программирование. Базовый курс C#: учебник для среднего профессионального образования / В. В. Подбельский. — Москва: Издательство Юрайт, 2020. — 369 с. — URL: <https://urait.ru/bcode/456697>.

6.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет».

Национальная электронная библиотека (НЭБ) www.rusneb.ru
 ELibrary.ru Научная электронная библиотека www.elibrary.ru
 Электронная библиотека Grebennikon.ru www.grebennikon.ru
 Cambridge University Press

6.3 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsu.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс
2. Гарант

7. Материально-техническое обеспечение дисциплины

№п/п	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа		
			Наименование ПО	Лицензия/сертификат/заказ	Дата лицензии
1.	Лаборатория информатики – ауд. № 203	1 компьютер преподавателя, 12 компьютеров обучающихся, маркерная доска, проектор	Windows 7 Microsoft office 2010 Pro Microsoft Visual Professional 2019 Mozilla Firefox 52.8.1 ESR Matlab Mathcad Education - University edition Kaspersky Endpoint	68526624 49420326 63202190 свободный доступ 647526 2996385 17E0-181226-094912-873-979	без даты 08.12.2011 без даты свободный доступ 14.06.2019 26.12.2018

			Security ZOOM	Лицензионное	
--	--	--	------------------	--------------	--

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.

- для глухих и слабослышащих: в печатной форме, в форме электронного документа.

- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA SE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;

- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемые эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1 Планы практических занятий

Тема 1 (4 ч.). Основы программирования на языке C# .

Задание:

1. Согласно заранее определенному варианту выполнить программирование операторов цикла и ветвления для решения алгебраической задачи;
2. Типизацию переменных задачи осуществить согласно типам данных C#;
3. Подготовить развернутый отчет о проделанной работе.

Тема 2 (4 ч.). Классы. Наследование. Многопоточное программирование C#.

Задание:

1. Согласно заранее определенному варианту выполнить объявление многомерного статического массива;
2. Выполнить поиск и перестановку минимального и максимального элемента.
3. Подготовить развернутый отчет о проделанной работе.

Тема 3 (4 ч.). Лямбда-функции. Введение в библиотеку C#.

Задание:

1. Согласно заранее определенному варианту выполнить программирование алгоритмов сортировки с использованием лямбда-функции C#;
2. Методы сортировки должны быть реализованы с использованием библиотек C#.lang ;
3. Подготовить развернутый отчет о проделанной работе.

Тема 4 (4 ч.). Чтение и запись данных в файлы. Автоматическое закрытие файла.

Задание:

1. Согласно заранее определенному варианту выполнить обработку C-подобных строковых данных и осуществить чтение/запись в файл;
2. Предусмотреть обработку текстовых(числовых) данных, хранящихся в файле, при помощи инструмента **struct**;
3. Подготовить развернутый отчет о проделанной работе.

Тема 5 (4 ч.). Чтение и запись данных в файлы. Автоматическое закрытие файла. Основы создания апплетов.

Задание:

1. Согласно заранее определенному варианту и используя принципы объектно-ориентированного программирования организовать создание объекта типа **class** в состав, которого входят методы сортировки динамически инициализированного двумерного массива, а также определены конструктор и деструктор класса;
2. Предусмотреть возможность создания дочерних классов на базе созданного родительского класса;
3. Подготовить развернутый отчет о проделанной работе.

Тема 6 (4 ч.). Использование стандартных интерфейсов

Задание:

1. Согласно заранее определенному варианту и используя принципы объектно-ориентированного программирования организовать создание стандартного интерфейса;
2. Предусмотреть возможность создания дочерних классов на базе созданного родительского класса;
3. Подготовить развернутый отчет о проделанной работе.

Тема 7 (4 ч.). Делегаты

Задание:

1. Согласно заранее определенному варианту и используя принципы объектно-ориентированного программирования организовать создание делегатов;
2. Предусмотреть возможность создания дочерних классов на базе созданного родительского класса;
3. Подготовить развернутый отчет о проделанной работе.

Тема 8 (4 ч.). Перегрузка операторов

Задание:

1. Согласно заранее определенному варианту и используя принципы объектно-ориентированного программирования организовать создание перегруженных операторов класса;
2. Предусмотреть возможность создания дочерних классов на базе созданного родительского класса;
3. Подготовить развернутый отчет о проделанной работе.

Тема 9 (8 ч.). Компоненты.

Задание:

1. Согласно заранее определенному варианту и используя принципы объектно-ориентированного программирования организовать создание собственных компонентов;
2. Предусмотреть возможность создания дочерних классов на базе созданного родительского класса;
3. Подготовить развернутый отчет о проделанной работе.

9.2 Методические рекомендации по подготовке письменных работ

Отчет по проделанной работе должен быть изложен с соблюдением правил грамматики русского и английского языков (в случаях необходимости). При этом отражаемые результаты работы должны быть информативными, тезисного порядка. В отчет входят следующие обязательные разделы:

1. Титульный лист с полным указанием ведомственной принадлежности, названия ВУЗа, института, факультета, кафедры. Кроме того, полное точное название практической работы, Ф.И.О. студента подготовившего отчет о результатах проделанной работы и Ф.И.О., должность, название кафедры преподавателя осуществляющего проверку и оценивание полученных результатов.
2. Содержание.
3. Введение.
4. Цели и задачи практической работы.
5. Методы и технологии, применяемые для решения поставленных задач оформленные в виде отдельных этапов работы.
6. Выводы по работе.
7. Приложения.

Оформление отчета выполняется с использованием компьютерной верстки **LaTeX**. Отчет сохраняется и представляет для проверки в виде отдельного **pdf** файла. В имени файла указывается фамилия студента и номер выполненной работы.

9.3 Рекомендации по написанию курсовой работы

При подготовке курсовой работы студент приобретает навыки сбора, обработки, анализа и изложения материала по изучаемому предмету.

Подготовка курсовой работы

1. Студент выбирает тему и согласовывает её с руководителем.
2. Затем студент приступает к сбору информации. Первоначальное представление о теме и структуре работы можно составить по учебникам, справочникам, монографиям, статьям в научных журналах. На этом этапе составляется и согласовывается с преподавателем план курсовой работы.
3. Собранный материал (выписки таблиц, графики) систематизируются в соответствии с планом. План может уточняться.
4. После этого пишется работа, обращается особое внимание на обработку информации, её анализ на основе последних данных, используя таблицы, графики, а также математический аппарат, если это целесообразно.
5. Составляется библиография (список использованных источников).

РЕКОМЕНДУЕМАЯ СТРУКТУРА РАБОТЫ

1. Титульный лист
2. Содержание (план работы)
3. Введение (предмет исследования, цель и её разбивка по задачам, как тема освещена в исследовательской литературе, наиболее актуальные и/или спорные аспекты)
4. 2-3 главы и параграфы (если разбивка на параграфы целесообразна). В них раскрывается суть раскрываемой темы. Материал излагается логично, последовательно. Данные приведенных таблиц, графиков затем анализируются в текстовой форме
5. Заключение (целесообразно написать в форме выводов)
6. Список использованной литературы и источников (библиография)

Оформление и научный аппарат

Работа открывается титульным листом. Титульный лист содержит информацию об учебном заведении, где выполнена работа, его подразделении, которое осуществляет руководство (кафедра), точную формулировку темы, сведения об авторе (фамилия, и. о., академическая группа), курс, факультет. Кроме того, должны быть указаны должность, ученая степень или ученое звание научного руководителя, его фамилия, и.о., место (Москва) и год написания реферата.

Научный аппарат курсовой работы должен отвечать требованиям, принятым в работах научного содержания.

Использование в тексте курсовой работы цитат, наиболее важных фактов и статистических показателей, особенно таблиц, а также графиков должно быть подкреплено ссылкой на источник либо внизу страницы, либо в конце работы в специальном разделе. Примеры: Сидоров И.Н. Сфера платных услуг в РФ. М., Студент, 2009, С.34; Иванова И.М., указ. соч., с. 45 (если ссылка давалась ранее). Для иностранных источников при неоднократных ссылках указывается: Op. cit и страница. При ссылках на статьи в периодических изданиях вначале указывается автор и название статьи, а затем в скобках – выходные данные собственно издания /журнала, газеты, сборника/.

Список литературы и источников приводится в конце курсовой работы и составляется либо в алфавитном порядке, либо с разбивкой по характеру изданий /официальные документы, монографии, статьи, статистические источники, периодика, Интернет.

Объём работы и сроки её сдачи

Объем курсовой работы – 25-30 стандартных страниц (в каждой около 30 строк по примерно 60 знаков). Вступление, заключение – по 2-3 страницы.

Срок сдачи в осеннем семестре – до 1 декабря, в весеннем семестре – до 1 мая. Оформленная работа сдается на кафедру.

Проверка и оценка работы

До начала зачетной сессии научный руководитель проверяет работу и выставляет оценку в форме зачета или дифференцированно по стобальной системе и пишет краткую рецензию. Доработка чистового варианта с целью устранения имеющихся недостатков в принципе возможна, если работа сдана в установленные деканатом сроки. В интересах успешной работы в будущем студенту рекомендуется встретиться с научным руководителем лично для более полного разбора курсовой работы.

Примерная тематика курсовых работ

1. Разработка прототипа электронной регистратуры
2. Разработка электронного теста "Правила дорожного движения"
3. Разработка программного обеспечения Screener
4. Разработка сетевого чата с применением службы WCF
5. Разработка игры "Ханойские башни"
6. Разработка автоматизированного рабочего места библиотекаря
7. Разработка автоматизированного рабочего места диспетчера такси
8. Разработка игровых модификаций на языке СИ #
9. Разработка системы для интерактивного обучения
10. Разработка программы помощника со встроенной поисковой системой
11. Разработка игры "Кроссворд"
12. Разработка системы "Домашняя бухгалтерия "
13. Разработка прототипа системы для ресторанного бизнеса
14. Разработка многопоточного приложения на языке СИ #
15. Разработка автоматизированного рабочего места диспетчера троллейбусного депо
16. Разработка интерактивной книги о вкусной и здоровой пище
17. Разработка игры "Пятнашки"
18. Реализация алгоритмов шифрования на языке СИ #
19. Разработка матричного калькулятора

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Цель дисциплины – приобретение знаний, навыков и умений в области технологий и методов программирования, а также освоение основных алгоритмов обработки и хранения больших данных.

Задачи дисциплины:

1. Изучение базовых принципов программирования на языке высокого уровня;
2. Дать широко применяемые технологии и методы программирования на языке C#, в том числе, специализированные методы обработки и хранения данных;
3. Изучение главных управляющих структур языка C#;
4. Изучение принципов объектно-ориентированного программирования;
5. Сформировать навыки и умения по разработке алгоритмов в задачах последовательного и объектно-ориентированного программирования;
6. Познакомить со стандартами языка C# и их отличительными особенностями;
7. Дать представления о процедуре отладки разрабатываемых приложений;
8. Сформировать представления о разработке эффективных приложений и обучить основным методам оптимизации;
9. Дать представление о разработке реляционных баз данных;
10. Обучить основам построения самообучающихся алгоритмов на языках высокого уровня.

В результате изучения дисциплины студент должен:

Знать: базовые принципы программирования на языке высокого уровня; главные управляющие структуры языка C#; принципы объектно-ориентированного программирования; алгоритмы отладки разрабатываемых приложений; методы разработки эффективных приложений и оптимизации по времени и памяти; основы построения самообучающихся алгоритмов; специализированные методы обработки и хранения данных.

Уметь: применять специализированные методы обработки и хранения данных на языке C#; разрабатывать алгоритмы для решения задач последовательного и объектно-ориентированного программирования; применять принципы объектно-ориентированного программирования для построения высоко эффективных приложений; разрабатывать простейшие модели реляционных баз данных на языке высокого уровня C#;

Владеть: базовыми принципами программирования на языке высокого уровня; методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение программы и используемой памяти компьютера; основными управляющими структурами языка C#; методами сортировки и хранения данных; методами и подходами к динамическому выделению памяти в пространстве памяти компьютера.