

УТВЕРЖДАЮ
 Декан факультета

« ____ » _____ 20__

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ

Направление/специальность подготовки	09.03.04 Программная инженерия
Специализация/профиль/программа подготовки	Разработка программно-информационных систем
Уровень высшего образования	Бакалавриат
Форма обучения	Очная
Факультет	И Робототехника и инновационная инженерия
Выпускающая кафедра	Н2 Программная инженерия и интеллектуальные системы
Кафедра-разработчик рабочей программы	Н2 Программная инженерия и интеллектуальные системы

КУРС	СЕМЕСТР	ОБЩАЯ ТРУДОЁМКОСТЬ (ЗАЧЕТНЫХ ЕДИНИЦ)	ЧАСЫ (по наличию видов занятий)									ВИД ПРОМЕЖУТОЧНОГО КОНТРОЛЯ
			ОБЩАЯ ТРУДОЁМКОСТЬ	АУДИТОРНЫЕ ЗАНЯТИЯ				САМОСТОЯТЕЛЬНАЯ РАБОТА				
				ВСЕГО	ЛЕКЦИИ	ЛАБОРАТОРНЫЙ ПРАКТИКУМ	ПРАКТИЧЕСКИЕ ЗАНЯТИЯ	ВСЕГО	КУРСОВОЙ ПРОЕКТ	КУРСОВАЯ РАБОТА	ДРУГИЕ ВИДЫ САМОСТ. РАБОТЫ	
1	2	4	144	51	34	0	17	93	0	0	93	диф. зач.
2	3	4	144	68	34	0	34	76	0	18	58	диф. зач.
ВСЕГО		8	288	119	68	0	51	169	0	18	151	

ЛИСТ СОГЛАСОВАНИЯ

РАБОЧАЯ ПРОГРАММА СОСТАВЛЕНА В СООТВЕТСТВИИ С ТРЕБОВАНИЯМИ ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА ВЫСШЕГО ОБРАЗОВАНИЯ (ФГОС ВО)

09.03.04 Программная инженерия

год набора группы: 2026

Программу составил:

Кафедра Н2 Программная инженерия и интеллектуальные системы
Вальштейн Константин Владимирович, старший преподаватель

Программа рассмотрена
на заседании кафедры-разработчика
рабочей программы **Н2 Программная инженерия и интеллектуальные системы**

Заведующий кафедрой Семенова Е.Г., д.т.н., проф.

Программа рассмотрена
на заседании выпускающей кафедры

Н2 Программная инженерия и интеллектуальные системы

Заведующий кафедрой Семенова Е.Г., д.т.н., проф.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ

Разделы рабочей программы

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО
3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ
4. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ
5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ
6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Приложения к рабочей программе дисциплины

- Приложение 1. Аннотация рабочей программы
- Приложение 2. Технологии и формы обучения
- Приложение 3. Фонды оценочных средств

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является формирование следующих компетенций:

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

ОПК-6 — Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

ПК-1.3 — Способен использовать различные технологии разработки программного обеспечения

ПК-93 — Способен генерировать новые идеи для решения задач цифровой экономики, абстрагироваться от стандартных моделей, перестраивать сложившиеся способы решения задач, выдвигать альтернативные варианты действий с целью выработки новых оптимальных алгоритмов

Формированию компетенций служит достижение следующих результатов образования:

ОПК-2

знания:

основы и техника обобщенного программирования;;;

умения:

выбирать способы создания программных продуктов исходя из доступных языковых и инструментальных средств;;;

навыки:

программирования статических и движущихся двумерных изображений на растровых устройствах отображения;;;

ОПК-6

знания:

понятия класса и объекта, понятия наследования, инкапсуляции и полиморфизма, понятие интерфейса класса и интерфейса библиотеки, виртуальные функции и их использование, базовые принципы функционирования программ с использованием библиотек семейств Simple DirectMedia Layer 2.x;;;

умения:

применять объектно-ориентированный подход при создании программных продуктов;;;

навыки:

программирования с использованием виртуальных функций и шаблонов;;;

программирования с использованием сторонних библиотек, расширяющих возможности базового языка программирования;;;

программирования статических и движущихся двумерных изображений на растровых устройствах отображения;;;

создания однооконных и многооконных Win32-приложений;;;

ПК-1.3

знания:

Способов практической реализации программных проектов на языке C++;;;

умения:

Разрабатывать структуру программных приложений;;;

навыки:

Использовать распространенные методики разработки программного обеспечения;;;

ПК-93

знания:

основы и техника объектно-ориентированного программирования;;;

умения:

выбирать способы создания программных продуктов исходя из доступных языковых и инструментальных средств;;;

навыки:

программирования с использованием сторонних библиотек, расширяющих возможности базового языка программирования;;;

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *09.03.04 Программная инженерия*.

Содержание дисциплины является логическим продолжением дисциплин: **ВВЕДЕНИЕ В ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ИНФОРМАТИКА: ОСНОВЫ ПРОГРАММИРОВАНИЯ.**

Содержание дисциплины является основой для освоения дисциплин: **СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.**

Предварительные компетенции, сформированные у обучающегося до начала изучения дисциплины:

- 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
- ОПК-3 — Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
- ОПК-6 — Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов
- ПК-94 — Способен к управлению информацией и данными, поиску источников информации и данных, восприятию, анализу, запоминанию и передаче информации с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 8 з.е., 288 ч.

3.1. Содержание (дидактика) дисциплины

КУРС	СЕМЕСТР	Наименование разделов и дидактических единиц	ВСЕГО	Аудиторные занятия в контактной форме			Самостоятельная работа студентов	Формируемая компетенция, %			
				ВСЕГО	Лекции	Практические занятия		ОПК-2	ОПК-6	ПК-1.3	ПК-93
1	2	Раздел 1. Основные понятия объектно-ориентированного программирования (ООП). 1.1. Парадигма ООП: инкапсуляция, наследование, полиморфизм. 1.2. Язык программирования (ЯП) C++ как язык, поддерживающий ООП. 1.3. Сравнение ЯП C++ и C. Указатели и ссылки. Возможность использования фрагментов текстов программ на C в программах на C++. 1.4. Обзор стандартной библиотеки C++. Пространства имен как механизм высокоуровневой инкапсуляции. Стандартные пространства имен. Заголовочные файлы стандартной библиотеки. 1.5. Поточковый ввод-вывод в C++. 1.6. Размещение динамических переменных. Операторы new, new[], delete, delete[]. 1.7. Многомерные динамические массивы. 1.8. Операции преобразование типа в C и C++. Операции static_cast, dynamic_cast, const_cast, reinterpret_cast.	9	4	4	0	5	10	10	10	10
1	2	Раздел 2. Стандартные и пользовательские типы данных в C++. Обработка исключений. Инкапсуляция и статический полиморфизм в C++. 2.1. Классы. Члены класса. Области видимости членов класса. 2.2. Конструкторы и деструкторы. Работа с экземпляром класса. Указатель this. 2.3. Исключения. Обработка исключений. Генерация исключений. Стандартные классы исключений. 2.4. Функции с параметрами по умолчанию. Перегрузка функций. 2.5. Конструктор копирования и оператор присваивания. Понятие статического полиморфизма. 2.6. Представление операций для классов. Операторные функции. 2.7. Дружественные функции и дружественные классы. 2.8. Статические члены класса.	26	9	6	3	17	5	5	5	5
1	2	Раздел 3. Наследование и динамический полиморфизм в C++. 3.1. Наследование. Иерархия классов. 3.2. Виртуальные функции. Понятие динамического полиморфизма. Виртуальные деструкторы. Оператор typeid. 3.3. Чисто виртуальные функции и абстрактные базовые классы. 3.4. Множественное наследование. 3.5. Указатели на компоненты класса. Доступ по указателю. 3.6. Объединения. 3.7. Ромбовидное наследование. Виртуальные базовые класс.	26	10	6	4	16	5	5	5	5
1	2	Раздел 4. Обобщенное программирование. Шаблоны функций и шаблоны классов. 4.1. Парадигма обобщенного программирования. C++ как язык поддерживающий обобщенное программирование. 4.2. Шаблоны функций. Оператор typename. 4.3. Шаблоны классов.	22	7	4	3	15	5	5	5	5
1	2	Раздел 5. Библиотека стандартных шаблонов (Standard Template Library, STL). 5.1. Итераторы STL. 5.2. Контейнеры STL. 5.3. Поток STL. 5.4. Функциональные объекты STL. 5.5. Алгоритмы STL.	22	7	4	3	15	5	5	5	5
1	2	Раздел 6. Кроссплатформенная библиотека для создания многооконных приложений SDL 2.x. 6.1. Модель множества окон отображения библиотеках SDL 2.x. Понятия окна, визуализатора и текстуры в SDL 2.x. Связь поверхности отображения и текстуры отображения SDL 2.x. 6.2. Модель обработки событий в SDL 2.x. Работа с очередью событий SDL 2.x. 6.3. Обзор дополнительных библиотек семейства Simple DirectMedia Layer 2.x. 6.4. Организация ввода данных в графическом режиме с использованием SDL 2.x. 6.5. Способы программного построения двумерных движущихся изображений в SDL 2.x.	26	10	6	4	16	5	5	5	5
1	2	Раздел 7. Расширение возможностей языка C++. 7.1. Основные версии стандартов C++ и широко применяемые расширения стандартов. 7.2. Библиотеки C++ не входящие в стандарт.	13	4	4	0	9	5	5	5	5
Всего за 2 семестр			144	51	34	17	93	40	40	40	40
2	3	Раздел 8. Общезыковая среда выполнения (CLR). Единая система типов .NET Framework. 8.1. Общезыковая среда выполнения (CLR). Базовая библиотека классов (BCL). Понятие сборки (assembly). Метаданные и манифест сборки. 8.2 Единая система типов .NET Framework. Типы-значения (value types) и ссылочные типы (reference types). Класс System.Object как общий базовый класс всех типов C#. Упаковка(boxing) и распаковка(unboxing) типов значений. Анонимные типы. Допустимые преобразования типов. Тип dynamic. 8.3 Структура программы на языке C#. Пространства имен.	12	4	2	2	8	8	8	8	8
2	3	Раздел 9. Массивы, строки, работа с файлами. 9.1 Массивы. Определение и инициализация. Допустимые приведения типов массивов. 9.2 Классы System.String и	14	6	2	4	8	6	6	6	6

		System.Text.StringBuilder для работы со строками. Классы для работы с файлами и каталогами, потоковый класс System.IO.FileStream, классы System.IO.BinaryReader и System.IO.BinaryWriter. 9.3 Понятие потока в контексте работы с файлами.										
2	3	Раздел 10. Пространства имен. Классы, структуры, интерфейсы. 10.1 Классы и структуры в C#. Сравнение написания классов на языке C++ и C#. Частичные классы. Частичные методы. Свойства и индексы. Автореализуемые свойства. Модификаторы ref и out. Методы с переменным числом параметров. Модификатор params. Перегрузка операторов. 10.2 Операторы as и is. Тип интерфейса - определение и реализация. Явная и неявная реализация интерфейса. Реализация интерфейсов и наследование.	20	12	8	4	8	6	6	6	6	6
2	3	Раздел 11. Исключения. Типы с явным освобождением ресурсов. Сборщик мусора. 11.1 Механизм исключений. Блоки catch и finally. Иерархия библиотечных классов-исключений. 11.2 Жизненный цикл объекта. Деструкторы и метод Finalize. Сборщик мусора. Типы с явным освобождением ресурсов. Сравнение подходов к решению задачи освобождения ресурсов в языках C# и C++.	13	4	2	2	9	6	6	6	6	6
2	3	Раздел 12. Типы-коллекции и универсальные (обобщенные) коллекции. 12.1 Пространство имен System.Collections. Итераторы. Блок итератора. Оператор yield. Интерфейсы IEnumerable и IEnumerator. Оператор foreach. Интерфейсы ICollection и IList. Класс ArrayList. Интерфейс IDictionary и класс Hashtable. Интерфейсы IComparable и IComparer. 12.2 Интерфейсы ICollection, IList и IDictionary. Классы List и Dictionary.	20	12	8	4	8	6	6	6	6	6
2	3	Раздел 13. Делегаты и события. 13.1 Тип delegate. Классы System.Delegate и System.MulticastDelegate. Анонимные методы. Обобщенные делегаты. 13.2 Определение и реализация событий. Свойства события (event properties). События и интерфейсы. Делегаты EventHandler и EventHandler. Тип System.EventArgs. Интерфейс System.ComponentModel.INotifyPropertyChanged.	17	8	4	4	9	6	6	6	6	6
2	3	Раздел 14. Дополнительные главы языка C#. 14.1 Механизм сериализации. Виды сериализации. Версия сборки. Сборки со строгим именем. Механизм отражения (reflection). Класс System.Type. Атрибуты. Определение пользовательских атрибутов. 14.2 Взаимодействие управляемого и неуправляемого кода. Сервис PInvoke. Атрибуты DllImport и MarshalAs. Маршalling типов-значений и ссылочных типов. 14.3 Глобализация и локализация приложения. Региональные настройки (culture).	16	8	4	4	8	6	6	6	6	6
2	3	Раздел 15. Разработка графических приложений на языке C#. 15.1 Интерфейс GDI+. Пространства имен System.Drawing, System.Drawing.Drawing2D, System.Drawing.Imaging, и System.Drawing.Text. 15.2 Класс Graphics и его методы. Отрисовка графических примитивов.	15	6	2	4	9	8	8	8	8	8
2	3	Раздел 16. Разработка оконных приложений на языке C#. 16.1 Технология Windows Forms. Структура проекта. Ресурсы проекта. Архитектурный шаблон MVC. Особенности создания приложения.	17	8	2	6	9	8	8	8	8	8
Всего за 3 семестр			144	68	34	34	76	60	60	60	60	60
Всего по дисциплине			288	119	68	51	169	100	100	100	100	100

3.2. Аудиторный практикум

№ п/п	Номер и наименование раздела дисциплины	Тема практического занятия	Объем, ауд. часов
1	Раздел 2. Стандартные и пользовательские типы данных в C++.	Стандартные и пользовательские типы данных в C++. Классы в C++: определение и использование.	1
2	Обработка исключений. Инкапсуляция и статический полиморфизм в C++.	Выполнение индивидуальной практической работы 2 (ИПР-1): описание класса с выводом отладочной информации во всех методах класса, написание программы тестирования всех методов класса с интерфейсом пользователя на основе консольного текстового меню.	2
3	Раздел 3. Наследование и динамический полиморфизм в C++.	Наследование. Иерархия классов. Виртуальные функции. Динамический полиморфизм в C++. Виртуальные деструкторы. Оператор typeid.	1
4		Чисто виртуальные функции и абстрактные базовые классы. Множественное наследование. Ромбовидное наследование. Виртуальные базовые классы.	1
5		Выполнение индивидуальной практической работы 2 (ИПР-2): разработка иерархии классов из заданного базового и заданных производных классов, с указанными в задании обязательными и дополнительными компонентами, написание программы тестирования всех методов всех класса в иерархии, с	2

		интерфейсом пользователя на основе консольного текстового меню.	
6		Шаблоны функций и шаблоны классов в C++. Оператор <code>typename</code> .	1
7	Раздел 4. Обобщенное программирование. Шаблоны функций и шаблоны классов.	Выполнение индивидуальной практической работы 3 (ИПР-3): разработка шаблона заданной функции и написание программы ее тестирования для указанных типов данных, разработка шаблона класса абстрактного типа данных и написание программы с интерфейсом пользователя на основе консольного текстового меню для проверки всех методов классов, создаваемых на основе шаблона для указанных типов данных.	2
8	Раздел 5. Библиотека стандартных шаблонов (Standard Template Library, STL).	Обобщенное программирование с использованием библиотеки стандартных шаблонов (Standard Template Library, STL) C++	1
9		Выполнение индивидуальной практической работы 4 (ИПР-4): использование стандартного шаблона и написание программы тестирования для указанных типов данных и написание программы с интерфейсом пользователя на основе консольного текстового меню для проверки всех методов классов, создаваемых на основе шаблона для указанных типов данных.	2
10	Раздел 6. Кроссплатформенная библиотека для создания многооконных приложений SDL 2.x.	Способы обеспечения синхронного и асинхронного выполнения частей программы средствами SDL 2.x. Таймеры SDL 2.x. Взаимосвязь механизма событий и таймеров в SDL 2.x.	1
11		Выполнение индивидуальной практической работы 5 (ИПР-5): написание программы построения графика функции и программы построения двумерного движущегося изображения	2
12		Способы программного построения двумерных движущихся изображений.	1
Всего за 2 семестр			17
13	Раздел 8. Общезыковая среда выполнения (CLR). Единая система типов .NET Framework.	Система типов C#, классы: определение и использование	2
14	Раздел 9. Массивы, строки, работа с файлами.	Работа с массивами, строками. Определение потоковых классов	2
15		Выполнение индивидуальной практической работы 6 (ИПР-6): работа со строками и файлами с использованием потоков	2
16	Раздел 10. Пространства имен. Классы, структуры, интерфейсы.	Сравнение написания классов в C++ и C#. Определение интерфейса. Использование интерфейса в механизме наследовани	2
17		Выполнение индивидуальной практической работы 7 (ИПР-7): разработка иерархии классов с использованием интерфейсов, абстрактных классов и других механизмов работы с наследованием	2
18	Раздел 11. Исключения. Типы с явным освобождением ресурсов. Сборщик мусора.	Работа с механизмом исключений. Жизненный цикл объекта. Работа со сборщиком мусора. Поколения жизни объекта	1
19		Выполнение индивидуальной практической работы 8 (ИПР-8): сравнение подходов к освобождению ресурсов в языках C++ и C#	1
20	Раздел 12. Типы-коллекции и универсальные (обобщенные) коллекции.	Понятие коллекции. Работа с коллекцией. Работа с итераторами. Универсальные интерфейсы коллекций.	4
21	Раздел 13. Делегаты и события.	Механизм делегатов. Реализация событий с использованием делегатов.	4
22	Раздел 14. Дополнительные главы языка C#.	Выполнение индивидуальной практической работы 9 (ИПР-9): изучение взаимодействия управляемого и неуправляемого кода	4
23	Раздел 15. Разработка	Отрисовка графических примитивов с использованием класса	4

	графических приложений на языке C#.	Graphics	
24	Раздел 16. Разработка оконных приложений на языке C#.	Способы создания оконных приложений с использованием технологии Windows Forms	6
Всего за 3 семестр			34

3.3. Самостоятельная работа студента (СРС)

№ п/п	Номер и наименование раздела дисциплины	Содержание учебного задания	Объем, часов
1	Раздел 1. Основные понятия объектно-ориентированного программирования (ООП).	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	5
2	Раздел 2. Стандартные и пользовательские типы данных в C++. Обработка исключений. Инкапсуляция и статический полиморфизм в C++.	Подготовка к практическим занятиям	4
3		Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	9
4		Оформление отчета по ИПР-1	4
5	Раздел 3. Наследование и динамический полиморфизм в C++.	Оформление отчета по ИПР-2	2
6		Подготовка к практическим занятиям	4
7		Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	10
8	Раздел 4. Обобщенное программирование. Шаблоны функций и шаблоны классов.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	9
9		Подготовка к практическим занятиям	3
10		Оформление отчета по ИПР-3	3
11	Раздел 5. Библиотека стандартных шаблонов (Standard Template Library, STL).	Оформление отчета по ИПР-4	2
12		Подготовка к практическим занятиям	4
13		Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	9
14	Раздел 6. Кроссплатформенная библиотека для создания многооконных приложений SDL 2.x.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	10
15		Подготовка к практическим занятиям	3
16		Оформление отчета по ИПР-5	3
17	Раздел 7. Расширение возможностей языка C++.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	9
Всего за 2 семестр			93
18	Раздел 8. Общезыковая среда выполнения (CLR). Единая система типов .NET Framework.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	7
19		Выполнение этапа 1 курсовой работы	1
20	Раздел 9. Массивы, строки, работа с файлами.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	4
21		Оформление отчета по ИПР-6	2
22		Выполнение этапа 2 курсовой	1

		работы	
23		Выполнение этапа 3 курсовой работы	1
24	Раздел 10. Пространства имен. Классы, структуры, интерфейсы.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	2
25		Выполнение этапа 5 курсовой работы	3
26		Оформление отчета по ИПР-7	1
27		Выполнение этапа 4 курсовой работы	2
28	Раздел 11. Исключения. Типы с явным освобождением ресурсов. Сборщик мусора.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	3
29		Оформление отчета по ИПР-8	1
30		Подготовка к практическим занятиям	1
31		Выполнение этапа 6 курсовой работы(часть 1)	4
32	Раздел 12. Типы-коллекции и универсальные (обобщенные) коллекции.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	2
33		Выполнение этапа 6 курсовой работы (часть 2)	6
34	Раздел 13. Делегаты и события.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	9
35	Раздел 14. Дополнительные главы языка C#.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	5
36		Оформление отчета по ИПР-9	3
37	Раздел 15. Разработка графических приложений на языке C#.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	9
38	Раздел 16. Разработка оконных приложений на языке C#.	Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	5
39		Оформление и подготовка к защите курсовой работы	4
Всего за 3 семестр			76

3.4. Курсовая работа

СОДЕРЖАНИЕ ЭТАПА	ПЕРИОД ИСПОЛНЕНИЯ (недели семестра)	ПЛАНИРУЕМОЕ ВРЕМЯ (час)
Этап 1. Анализ задачи, формулирование ограничений на конкретную реализацию. Описание предполагаемого процесса взаимодействия пользователя с программой.	1 - 2	1
Этап 2. Объектно-ориентированный анализ и проектирование: выявление классов, объектов и их отношений	3 - 4	1
Этап 3. Выполнение программной реализации основных классов, написание демонстрационно-тестирующей программы.	5 - 6	1
Этап 4. Программная реализация каркаса графического интерфейса с использованием выбранного языка программирования и технологий	7 - 9	2
Этап 5. Программная реализация необходимого взаимодействия средствами выбранной технологии	10 - 11	3

Этап 6. Подготовка окончательного варианта программы, включая встроенные справочные материалы. Подготовка документации для пользователя	12 - 16	10
Всего за 3 семестр		18

4. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

СЕМЕСТР	НЕДЕЛИ СЕМЕСТРА																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2				Отч. по ПЗ		ДР		Отч. по ПЗ		ДР	Отч. по ПЗ			Отч. по ПЗ		ДР	Отч. по ПЗ, диф. зач.
3				Отч. по ПЗ		ДР		Отч. по ПЗ		ДР		Отч. по ПЗ				ДР	Отч. по ПЗ, КР, диф. зач.

Условные обозначения:

- ДР – диагностическая работа;
- Отч. по ПЗ – отчет по практическому заданию;
- КР – курсовая работа;
- диф. зач. – дифференцированный зачет;
- диф. зач. – дифференцированный зачет.

Текущий контроль успеваемости студентов проводится в дискретные временные интервалы в следующих формах:

- диагностическая работа;
- отчет по практическому заданию;
- курсовая работа.

Промежуточная аттестация проводится в формах:

- дифференцированный зачет.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Основная литература по дисциплине:

1. А. Н. Гуцин. . Применение библиотеки SDL для разработки программ на языке С. СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014, 50 экз.
2. А.А. Бармина, К. В. Вальштейн. . Программирование на языке С#. СПб.: Изд-во БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2022, 290 экз.
3. В. И. Юров. . Assembler. М.: Питер, 2006, 59 экз.
4. В. И. Юров. . Assembler. СПб.: Питер, 2010, эл. рес.
5. Дж. Бишоп, Н. Хорспул. . С# в кратком изложении. М.: БИНОМ. Лаборатория знаний, 2005, 5 экз.
6. И. Г. Головин, И. А. Волкова. . Языки и методы программирования. М.: Академия, 2016, 50 экз.
7. Н. А. Тюкачев, В. Г. Хлебостроев. . С#. Программирование 2D и 3D векторной графики. Санкт-Петербург: Лань, 2022, эл. рес.
8. О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня. СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014, эл. рес.
9. О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня. СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014, 60 экз.
10. С. С. Сосинская. . Использование языка С# в различных информационных технологиях. Старый Оскол: ТНТ, 2020, эл. рес.
11. Т. А. Павловская. . C/C++. Программирование на языке высокого уровня . Санкт-Петербург: Питер, 2021, эл. рес.
12. Э. Троелсен. . С# и платформа .NET. СПб.: Питер, 2005, 20 экз.
13. Ю. А. Щупак. . Win32 API. Разработка приложений для Windows. СПб.: Питер, 2008, 48 экз.

5.2. Дополнительная литература по дисциплине:

не требуется.

5.3. Периодические издания:

не требуются.

5.4. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины, электронные библиотечные системы:

1. <https://urait.ru/> — Образовательная платформа «Юрайт». Для вузов и ссузов.;;
2. <https://e.lanbook.com/> — ЭБС Лань;;
3. <http://sourceforge.net/projects/orwelldevcpp/> — Dev-C++ download | SourceForge.net;;
4. <http://www.codeblocks.org/> — Code::Blocks - Code::Blocks;;
5. <http://scholar.google.ru/> — Академия Google;;
6. <http://www.open-std.org/jtc1/sc22/wg21/> — ISO/IEC JTC1/SC22/WG21 - The C++ Standards Committee - ISO C++;;
7. <https://docs.microsoft.com/> — Developer tools, technical documentation and coding examples | Microsoft Docs;;

Современные профессиональные базы данных:

1. <https://rusneb.ru> – Национальная электронная библиотека (НЭБ);
2. <https://cyberleninka.ru/> - Научная электронная библиотека «Киберленинка»;
<http://www.rfbr.ru/rffi/ru/library> - Полнотекстовая электронная библиотека Российского фонда фундаментальных исследований.

Информационные справочные системы:

1. Техэксперт – Информационный портал технического регулирования: Нормы, правила, стандарты РФ;
2. http://library.voenmeh.ru/jirbis2/index.php?option=com_irbis&view=irbis&Itemid=457 - БД ГОСТов собственной генерации БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова;
3. <http://www.consultant.ru/> - КонсультантПлюс- информационный портал правовой информации.

5.5. Программное обеспечение:

1. Code::Blocks;
2. Linux;
3. Microsoft Visual Studio Community.

5.6. Информационные технологии:

взаимодействие с обучающимися посредством ЭИОС Moodle БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Лекционные занятия:

специализированные требования по оборудованию отсутствуют; аудитория с посадочными местами по количеству студентов; доска.

6.2. Практические занятия:

1. Проектор;
2. Code::Blocks;
3. Linux;
4. Microsoft Visual Studio Community.

6.3. Прочее:

1. рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
2. рабочие места студентов, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде.

Аннотация рабочей программы

Дисциплина **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ** является дисциплиной **обязательной части блока 1** программы подготовки по направлению *09.03.04 Программная инженерия*. Дисциплина реализуется на факультете *И Робототехника и инновационная инженерия* БГТУ "ВОЕНМЕХ" им. Д.Ф. Устинова кафедрой *И2 Программная инженерия и интеллектуальные системы*.

Дисциплина нацелена на формирование *компетенций*:

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;

ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;

ПК-1.3 Способен использовать различные технологии разработки программного обеспечения;

ПК-93 Способен генерировать новые идеи для решения задач цифровой экономики, абстрагироваться от стандартных моделей, перестраивать сложившиеся способы решения задач, выдвигать альтернативные варианты действий с целью выработки новых оптимальных алгоритмов.

Содержание дисциплины охватывает круг вопросов, связанных с использованием языков программирования высокого уровня при разработки программного обеспечения. Основное внимание уделяется парадигмам объектно-ориентированного программирования и обобщенного программирования, а также расширения языков программирования высокого уровня сторонними библиотеками, рассматриваются вопросы разработки программ с графическим пользовательским интерфейсом и взаимодействия программ с программными интерфейсами операционных систем. Также приводятся общие сведения о процессах, потоках, синхронном и асинхронном взаимодействии программ и их частей. В качестве основы для практических примеров рассматриваются язык программирования C++, библиотека стандартных шаблонов C++ STL, библиотека 2.x (SDL 2.x). Так же в качестве основы для практических примеров рассматриваются язык программирования C# и основанные на нем интерфейсы программирования приложений Windows Forms и WPF, а также внутренние библиотеки программной платформы .NET Core.

Программой дисциплины предусмотрены следующие **виды контроля**:

Текущий контроль успеваемости студентов проводится в дискретные временные интервалы в следующих формах:

- диагностическая работа;
- отчет по практическому заданию;
- курсовая работа.

Промежуточная аттестация проводится в формах:

- дифференцированный зачет.

Общая трудоемкость освоения дисциплины составляет **8 з.е., 288 ч.** Программой дисциплины предусмотрены лекционные занятия (**68 ч.**), практические занятия (**51 ч.**), самостоятельная работа студента (**169 ч.**).

ТЕХНОЛОГИИ И ФОРМЫ ОБУЧЕНИЯ

Рекомендации по освоению дисциплины для студента

Трудоемкость освоения дисциплины составляет 288 ч., из них 119 ч. аудиторных занятий, и 169 ч., отведенных на самостоятельную работу студента.

Рекомендации по распределению учебного времени по видам самостоятельной работы и разделам дисциплины приведены в таблице.

Контроль освоения дисциплины производится в соответствии с Положением о текущем, рубежном контроле успеваемости и промежуточной аттестации обучающихся.

Формы контроля и критерии оценивания приведены в приложении 3 к Рабочей программе.

Наименование работы	Рекомендуемая литература	Трудоемкость, час.
Раздел 1. Основные понятия объектно-ориентированного программирования (ООП).		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (1) Т. А. Павловская. . С/С++. Программирование на языке высокого уровня : Санкт-Петербург: Питер, 2021 (1) О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1) О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1)	5
Итого по разделу 1		5
Раздел 2. Стандартные и пользовательские типы данных в С++. Обработка исключений. Инкапсуляция и статический полиморфизм в С++.		
Подготовка к практическим занятиям	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (2)	4
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (2) Т. А. Павловская. . С/С++. Программирование на языке высокого уровня : Санкт-Петербург: Питер, 2021 (2) О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (2)	9
Оформление отчета по ИПР-1	СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (2)	4
Итого по разделу 2		17
Раздел 3. Наследование и динамический полиморфизм в С++.		
Оформление отчета по ИПР-2	В. И. Юров. . Assembler: СПб.: Питер, 2010 (3)	2
Подготовка к практическим занятиям	В. И. Юров. . Assembler: М.: Питер, 2006 (3) Ю. А. Щупак. . Win32 API. Разработка приложений для Windows: СПб.: Питер, 2008 (5)	4
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	Т. А. Павловская. . С/С++. Программирование на языке высокого уровня : Санкт-Петербург: Питер, 2021 (1-3)	10
Итого по разделу 3		16

Раздел 4. Обобщенное программирование. Шаблоны функций и шаблоны классов.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1-3) Т. А. Павловская. . С/С++. Программирование на языке высокого уровня : Санкт-Петербург: Питер, 2021 (1-3)	9
Подготовка к практическим занятиям	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1-3)	3
Оформление отчета по ИПР-3	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (1-3)	3
Итого по разделу 4		15
Раздел 5. Библиотека стандартных шаблонов (Standard Template Library, STL).		
Оформление отчета по ИПР-4	Т. А. Павловская. . С/С++. Программирование на языке высокого уровня : Санкт-Петербург: Питер, 2021 (5)	2
Подготовка к практическим занятиям	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (5)	4
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (5) И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (1-3)	9
Итого по разделу 5		15
Раздел 6. Кроссплатформенная библиотека для создания многооконных приложений SDL 2.x.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (1-3) Т. А. Павловская. . С/С++. Программирование на языке высокого уровня : Санкт-Петербург: Питер, 2021 (1-4) О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1-4)	10
Подготовка к практическим занятиям	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1-4)	3
Оформление отчета по ИПР-5	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1-4)	3
Итого по разделу 6		16
Раздел 7. Расширение возможностей языка С++.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (5) А. Н. Гуцин. . Применение библиотеки SDL для разработки программ на языке С: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (1-4) Т. А. Павловская. . С/С++. Программирование на языке высокого уровня : Санкт-Петербург: Питер, 2021 (5) О. В. Арипова, А. Н. Гуцин, О. А. Палехова. . Программирование на языке высокого уровня: СПб.БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2014 (5)	9
Итого по разделу 7		9

Раздел 8. Общеязыковая среда выполнения (CLR). Единая система типов .NET Framework.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (1-2) Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (1-2) И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (1)	7
Выполнение этапа 1 курсовой работы	С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (1)	1
Итого по разделу 8		8
Раздел 9. Массивы, строки, работа с файлами.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (3) Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (3)	4
Оформление отчета по ИПР-6	Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (3)	2
Выполнение этапа 2 курсовой работы	С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (1-2)	1
Выполнение этапа 3 курсовой работы		1
Итого по разделу 9		8
Раздел 10. Пространства имен. Классы, структуры, интерфейсы.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (5) С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (3)	2
Выполнение этапа 5 курсовой работы	Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (3) А.А. Бармина, К. В. Вальштейн. . Программирование на языке С#: СПб.: Изд-во БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2022 (1)	3
Оформление отчета по ИПР-7		1
Выполнение этапа 4 курсовой работы	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (1)	2
Итого по разделу 10		8
Раздел 11. Исключения. Типы с явным освобождением ресурсов. Сборщик мусора.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (5) Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (3)	3
Оформление отчета по ИПР-8	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (1)	1
Подготовка к практическим занятиям	С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (2)	1
Выполнение этапа 6 курсовой работы(часть 1)		4
Итого по разделу 11		9
Раздел 12. Типы-коллекции и универсальные (обобщенные) коллекции.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (8) Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (7-8) И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (4) А.А. Бармина, К. В. Вальштейн. . Программирование на языке С#: СПб.: Изд-во БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2022 (2)	2
Выполнение этапа 6 курсовой работы (часть 2)	С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (4-5)	6

Итого по разделу 12		8
Раздел 13. Делегаты и события.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (10) И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (3) Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (5) С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (3)	9
Итого по разделу 13		9
Раздел 14. Дополнительные главы языка С#.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (3) А.А. Бармина, К. В. Вальштейн. . Программирование на языке С#: СПб.: Изд-во БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2022 (3) Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (7) Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (20) С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (7)	5
Оформление отчета по ИПР-9	С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (7)	3
Итого по разделу 14		8
Раздел 15. Разработка графических приложений на языке С#.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (17) Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (5) С. С. Сосинская. . Использование языка С# в различных информационных технологиях: Старый Оскол: ТНТ, 2020 (5) А.А. Бармина, К. В. Вальштейн. . Программирование на языке С#: СПб.: Изд-во БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2022 (4) И. Г. Головин, И. А. Волкова . . Языки и методы программирования: М.: Академия, 2016 (10)	9
Итого по разделу 15		9
Раздел 16. Разработка оконных приложений на языке С#.		
Изучение предусмотренных программой дидактических единиц по рекомендуемой литературе	Дж. Бишоп, Н. Хорспул. . С# в кратком изложении: М.: БИНОМ. Лаборатория знаний, 2005 (1-5) Э. Троелсен. . С# и платформа .NET: СПб.: Питер, 2005 (18) А.А. Бармина, К. В. Вальштейн. . Программирование на языке С#: СПб.: Изд-во БГТУ "ВОЕНМЕХ" им. Д. Ф. Устинова, 2022 (4)	5
Оформление и подготовка к защите курсовой работы	Н. А. Тюкачёв, В. Г. Хлебостроев. . С#. Программирование 2D и 3D векторной графики: Санкт-Петербург: Лань, 2022 (1-3)	4
Итого по разделу 16		9

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств, позволяющие оценить результаты обучения по данной дисциплине, включают в себя:

- диагностическая работа
- отчет по практическому заданию;
- курсовая работа;
- дифференцированный зачет;
- дифференцированный зачет.

Критерии оценивания

Диагностическая работа

Диагностическая работа проводится в форме теста в ЭИОС Moodle:

- при правильном ответе менее чем на 60% вопросов - не аттестация;
- при правильном ответе на 60% вопросов и более - аттестация.

Отчет по практическому заданию

Во втором семестре необходимо выполнить пять ИПР. В третьем - четыре ИПР.

По всем ИПР необходимо успешное выполнение требования общей и вариативной части задания, включая предъявление в работе самостоятельно написанных соответствующих программ, в том числе на основе указанных примеров, если это предусмотрено заданием.

Отчет по ИПР:

Дополнительно к представлению всех результатов выполнения ИПР в электронной форме предусмотрено оформление отчетов, состоящих из титульного листа, вариативной части задания и основных результатов работы программы, а также наиболее соответствующих теме задания фрагментов разработанных программ.

Защита ИПР:

Защита ИПР предусматривает обсуждение порядка решения предусмотренных его тематикой задач, включая проверку усвоения студентом соответствующих сведений из теории и степени самостоятельности при написании предъявляемых программ.

Подробные критерии оценивания ИПР указаны в технологической карте дисциплины, размещённой в курсе в ЭИОС.

Курсовая работа

Выполненные курсовые работы представляются в электронной форме в виде подготовленных к сборке исходных текстов и полностью готовой к выполнению программы для тестирования преподавателем и электронной версии пояснительной записки, оформленной в соответствии с Положением по содержанию, оформлению, организации выполнения и защиты курсовых проектов и курсовых работ. При успешном тестировании программы и проверке соответствия пояснительной записки требованиям Положения и требованиям задания на данную курсовую работу, дается разрешение на ее печать без исходных текстов программ (они заменяются на «приложение в электронной форме»). При наличии распечатанной пояснительной записки студент допускается к защите КРР.

Критерии оценивания:

Курсовая работа допускается к защите при следующих условиях:

- предъявляемая программа работоспособна;
- программа выполнена соответствии с заданием;
- электронная и печатная версии пояснительной записки соответствуют установленным требованиям.

Оценка написанной КР:

- Работа выполнена, но не соответствует теме либо не использованы требуемые технологии, либо не реализованы все заявленные требования – 3 балла
- Работа выполнена в соответствии с темой, реализовано 85% заявленных возможностей, пользовательский интерфейс не предусматривает проверки ввода и не исправлены технические ошибки

(утечки памяти, программа некорректно завершает работу) - 6 баллов

- Работа выполнена, реализованы все заявленные возможности, пользовательский интерфейс содержит фразы на английском языке, отсутствует инструкция для пользователя по работе с программой - 9 баллов

- Работа выполнена, реализованы все возможности, ошибок в работе и интерфейсе не выявлено - 10 баллов

Оценка содержания пояснительной записки к курсовой работе:

- Пояснительная записка не содержит описания структуры разработанной программы, тестирование программы не произведено, продемонстрирован исключительно пользовательский интерфейс – 2 балла

- Пояснительная записка содержит описание структуры разработанной программы без использования диаграмм классов, тестирование программы не произведено, продемонстрирован пользовательский интерфейс и результаты работы программы – 3 балла

- Структура программы описана минимум одной диаграммой классов, описан базовый процесс тестирования, записка имеет четкую структуру в виде выделенных разделов и подразделов – 4 балла

- Структура программы описана диаграммами нескольких типов, полностью описан процесс тестирования, записка имеет четкую структуру в виде выделенных разделов и подразделов - 5 баллов

Оценка оформления, стиля пояснительной записки

- Пояснительная записка оформлена с нарушениями, язык работы не соответствует научному стилю, некорректно оформленные заимствования, некорректно оформлен список источников – 2 балла

- Пояснительная записка оформлена с нарушениями, язык работы не соответствует научному стилю, есть замечания к оформлению списка источников – 3 балла

- Есть отдельные замечания к оформлению и стилю изложения, оформлению списка источников – 4 балла

- Нет замечаний к оформлению и стилю изложения, оформлению списка источников – 5 баллов

Максимальное количество баллов – 20

Оценка «отлично» - 17-20 баллов

Оценка «хорошо» - 13-16 баллов

Оценка «удовлетворительно» - 10-12 баллов

Оценка «не защитил» - меньше 10 или работа не была предъявлена

Дифференцированный зачет (семестр 2)

Дифференцированный зачет проводится в виде электронного тестирования в ЭИОС.

В тесте 30 вопросов с максимальным баллом 30.

Шкала оценивания:

0 - 15 баллов - неудовлетворительно.

16 - 20 баллов - удовлетворительно.

21 - 25 баллов - хорошо.

26 - 30 баллов - отлично.

На тест дается 90 минут. Можно улучшить свой результат каждые 3 суток.

При выполнении и защите всех ПЗ до начала промежуточной аттестации предусмотрено повышение оценки на одну ступень начиная с оценки "зачтено-удовлетворительно".

При выполнении и защите всех ПЗ в усложненном варианте предусмотрена оценка "зачтено-отлично" по результатам работы в семестре.

Также предусмотрено получение оценки согласно набранным во время семестра баллам, согласно размещенной в курсе в ЭОИС технологической карте.

Дифференцированный зачет (семестр 3)

Дифференцированный зачет проводится в виде электронного тестирования в ЭИОС.

В тесте 30 вопросов с максимальным баллом 30.

Шкала оценивания:

0 - 15 баллов - не зачтено.

16 - 20 баллов - удовлетворительно.

21 - 25 баллов - хорошо.

26 - 30 баллов - отлично.

На тест дается 90 минут. Можно улучшить свой результат каждые 3 суток.

При выполнении и защите всех ПЗ до начала промежуточной аттестации предусмотрено повышение оценки на одну ступень начиная с оценки "зачтено-удовлетворительно".

При выполнении и защите всех ПЗ в усложненном варианте предусмотрена оценка "зачтено-отлично" по результатам работы в семестре.

Также предусмотрено получение оценки согласно набранным во время семестра баллам, согласно размещённой в курсе в ЭОИС технологической карте.

КУРС	СЕМЕСТР	Наименование разделов и дидактических единиц	ВСЕГО	Аудиторные занятия в контактной форме			Самостоятельная работа студентов	Формируемая компетенция, %				НАИМЕНОВАНИЕ ОЦЕНОЧНОГО СРЕДСТВА
				ВСЕГО	Лекции	Практические занятия		ОПК-2	ОПК-6	ПК-1.3	ПК-93	
1	2	Раздел 1. Основные понятия объектно-ориентированного программирования (ООП).	9	4	4	0	5	10	10	10	10	Отчет по практическому заданию
1	2	Раздел 2. Стандартные и пользовательские типы данных в C++. Обработка исключений. Инкапсуляция и статический полиморфизм в C++.	26	9	6	3	17	5	5	5	5	Отчет по практическому заданию
1	2	Раздел 3. Наследование и динамический полиморфизм в C++.	26	10	6	4	16	5	5	5	5	Отчет по практическому заданию
1	2	Раздел 4. Обобщенное программирование. Шаблоны функций и шаблоны классов.	22	7	4	3	15	5	5	5	5	Отчет по практическому заданию
1	2	Раздел 5. Библиотека стандартных шаблонов (Standard Template Library, STL).	22	7	4	3	15	5	5	5	5	Отчет по практическому заданию
1	2	Раздел 6. Кроссплатформенная библиотека для создания многооконных приложений SDL 2.x.	26	10	6	4	16	5	5	5	5	Отчет по практическому заданию
1	2	Раздел 7. Расширение возможностей языка C++.	13	4	4	0	9	5	5	5	5	Отчет по практическому заданию
Всего за 2 семестр			144	51	34	17	93	40	40	40	40	
2	3	Раздел 8. Общеязыковая среда выполнения (CLR). Единая система типов .NET Framework.	12	4	2	2	8	8	8	8	8	Отчет по практическому заданию
2	3	Раздел 9. Массивы, строки, работа с файлами.	14	6	2	4	8	6	6	6	6	Отчет по практическому заданию

2	3	Раздел 10. Пространства имен. Классы, структуры, интерфейсы.	20	12	8	4	8	6	6	6	6	Отчет по практическому заданию
2	3	Раздел 11. Исключения. Типы с явным освобождением ресурсов. Сборщик мусора.	13	4	2	2	9	6	6	6	6	Отчет по практическому заданию
2	3	Раздел 12. Типы- коллекции и универсальные (обобщенные) коллекции.	20	12	8	4	8	6	6	6	6	Отчет по практическому заданию
2	3	Раздел 13. Делегаты и события.	17	8	4	4	9	6	6	6	6	Отчет по практическому заданию
2	3	Раздел 14. Дополнительные главы языка C#.	16	8	4	4	8	6	6	6	6	Отчет по практическому заданию
2	3	Раздел 15. Разработка графических приложений на языке C#.	15	6	2	4	9	8	8	8	8	Отчет по практическому заданию
2	3	Раздел 16. Разработка оконных приложений на языке C#.	17	8	2	6	9	8	8	8	8	Курсовая работа
Всего за 3 семестр			144	68	34	34	76	60	60	60	60	
Всего по дисциплине			288	119	68	51	169	100	100	100	100	

Оценочные материалы по дисциплине ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ

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

№ 1 Прочитайте текст и запишите развернутый обоснованный ответ
Почему при перегрузке инжектора в поток (операция <<) в С++ необходимо использовать дружественную функцию?

№ 2 Прочитайте текст и установите последовательность
Установите последовательность этапов обработки исключения в С++:

Варианты:

Генерация исключения (throw)

Перехват исключения (catch)

Попытка выполнения кода (try)

Завершение обработки

№ 3 Прочитайте текст и установите последовательность
Расположите этапы выполнения LINQ-запроса в С#:

Варианты:

Определение источника данных

Написание запроса

Выполнение запроса

Обработка результата

№ 4 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа
Какое из следующих утверждений верно в С#?

Struct – это value type, Class – reference type

Struct – это reference type, Class –value type

И Struct и Class – это reference type

Тип (value или reference) зависит от типа полей

№ 5 Прочитайте текст и запишите развернутый обоснованный ответ
Все ли операторы языка С++ могут быть перегружены? Поясните ответ.

№ 6 Прочитайте текст и установите соответствие
Сопоставьте следующие директивы С# и их описание

1 #if

2 #ifdef

3 #define

А Проверка определения идентификатора

Б Отсутствует в языке С#

В Определяет конкретный идентификатор

Г Указывает блок программы для возможности скрыть его

№ 7 Прочитайте текст и установите соответствие

Соотнесите типы из WinAPI с типами, которые эквивалентны им в C# при маршаллинге

1 BOOL
2 LPSTR
3 LPDWORD

A bool
Б sbyte*
В uint*
Г char*

№ 8 Прочитайте текст и установите последовательность

Установите порядок этапов компиляции C++-программы:

Варианты:

Линковка

Препроцессинг

Компиляция

Исполнение

№ 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Для доступа к элементам объекта в C++ используются

при обращении через имя объекта – точка, при обращении через указатель – операция «->»
при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «точка»
при обращении через имя объекта – точка, при обращении через указатель – два двоеточия
при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»

№ 10 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Выберите верное утверждение относительно языка C++

И конструкторов, и деструкторов может быть много, при этом количество конструкторов должно быть не меньше количества деструкторов.

Конструкторов может быть много, а деструктор - только один.

В классе может быть один конструктор и несколько деструкторов.

В классе может быть только один конструктор и один деструктор.

№ 11 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Выберите классы потоков, объекты которых можно использовать для сериализации объектов.

Для сериализации не требуется поток

FileStream

MemoryStream

XMLStream

Stream

№ 12 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Как можно создать экземпляр типа в C#? Выберите все возможные варианты из представленных.

Activator.CreateInstance

AppDomain.CurrentDomain.CreateInstance

type.InvokeMember

Type.GetType

№ 13 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие методы создаются автоматически при объявлении класса в C++?

конструктор по умолчанию

конструктор копирования

деструктор

оператор присваивания

деструктр с параметрами

ОПК-6 - Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

№ 1 Прочитайте текст и установите последовательность

Установите порядок инициализации и вызова шаблонной функции в C++:

Варианты:

Определение шаблона

Инстанцирование с конкретным типом

Компиляция конкретного экземпляра

Вызов функции

№ 2 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какое минимальное количество конструкторов и деструкторов может быть в классе в C++?

2 конструктора, 1 деструктор

1 конструктор, 1 деструктор

3 конструктора, 1 деструктор

2 конструктора, 2 деструктора

№ 3 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Каково поведение Equals (object objA, object objB) в C#?

Если оба объекта не представляют одну и ту же ссылку на объект и ни один из них не имеет значения NULL, то вызывает objA.Equals (objB) и возвращает результат.

Если оба объекта не представляют одну и ту же ссылку на объект и ни один из них не имеет значения NULL, то вызывает objB.Equals (objA) и возвращает результат.

Является псевдонимом для objA.Equals (objB)

Сравнивает типы объектов, проверяет ссылочную целостность и вызывает objA.Equals (objB)

№ 4 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Что сделает программа C#, выполнив следующий код: Console.WriteLine(«Hello, World!»)?

Удалит все значения с Hello, World!

Напишет Hello, World!

Вырежет слово Hello, World! из всего текста

Напишет на новой строчке Hello, World!

№ 5 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Выберите варианты, при которых наступает сборка мусора

Принудительный запуск GC.Collect()

ОС сообщает о нехватке памяти

Завершение работы CLR

Вызов у объекта метода Stop()

№ 6 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор

ответов

Компилятор "разворачивает" делегат в класс. Выберите методы, которые обязательно присутствуют в этом классе в C#.

Invoke

BeginInvoke

EndInvoke

Конструктор класса

- № 7 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Выберите действия, необходимые для сериализации объекта.

Создание или получение потока

Создание форматтера

Вызов метода для сериализации

Закрытие потока

Закрытие форматтера

- № 8 Прочитайте текст и запишите развернутый обоснованный ответ

Пусть имеется некоторый класс A. В программе на языке C++ есть строка:

```
extern A a;
```

Что происходит в данной строке?

- № 9 Прочитайте текст и запишите развернутый обоснованный ответ

Как может быть создан объект некоторого класса в C++?

- № 10 Прочитайте текст и установите соответствие

Сопоставьте элементы C++ с их описанием

Элемент

A. new

B. delete

C. #include

D. std::vector

Варианты:

Заголовочный файл включается в программу

Динамическое выделение памяти

Удаление динамически выделенной памяти

Стандартный контейнер для хранения последовательности элементов

Пространство имён для классов и функций

Условный компиляторный оператор

- № 11 Прочитайте текст и установите соответствие

Вопрос 2. Сопоставьте директивы препроцессора C++ с их назначением.

Директива

A. #define

B. #ifdef

C. #endif

D. #pragma

E. #undef

Варианты:

Завершает условную компиляцию

Создаёт макроопределение

Проверяет наличие макроопределения

Удаляет макроопределение

Указание компилятору особых настроек

Подключает файл в код

№ 12 Прочитайте текст и установите последовательность

Упорядочите этапы разработки консольного приложения на C#:

Варианты:

Написание кода

Компиляция в IL

Исполнение через JIT

Загрузка CLR

ПК-1.3 - Способен использовать различные технологии разработки программного обеспечения

№ 1 Прочитайте текст и установите последовательность

Упорядочите шаги при работе с динамической памятью в C++:

Варианты:

Выделение памяти

Проверка корректности выделения

Использование

Освобождение памяти

№ 2 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Пусть имеется некоторый класс A и объект a:

```
class A {  
    double Num;  
public:  
    double getNum();  
    void setNum(double d);  
} a;
```

Где-то в программе есть строка:

```
a.setNum(1.57);
```

Что происходит в данной строке в C++?

Вызов метода setNum() для объекта a непосредственно

Вызов метода setNum() для объекта a по указателю

Вызов метода setNum() для объекта a по ссылке
Вызов метода setNum() для объекта a через точку

- № 3 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа
Какого спецификатора доступа в C++ нет?
- public
 - internal
 - private
 - protected
- № 4 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа
Какой тип переменной используется в коде в C#: int value = 42
Знаковое 32-бит целое
Знаковое
8-бит целое
1 байт*
Знаковое
64-бит целое
- № 5 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов
Выберите принципы ООП
Инкапсуляция
Полиморфизм
Наследование
Обфускация
Рефлексия
Структуризация
- № 6 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов
Что из перечисленного нельзя объявлять с модификатором static в C#?
Конструктор
Свойство
Финализатор
Индексатор
- № 7 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов
Выберите допустимые типы конструкторов в C#
Конструктор по умолчанию
Конструктор копирования
Конструктор с параметрами
Статический конструктор
- № 8 Прочитайте текст и запишите развернутый обоснованный ответ
В каких случаях необходимо перегружать оператор присваивания в C++?
- № 9 Прочитайте текст и запишите развернутый обоснованный ответ
Какими способами можно обратиться к компонентам объекта некоторого класса в C++?
- № 10 Прочитайте текст и установите соответствие
Сопоставьте типы данных C# с их аналогами в C++.
- C# тип
 - A. int
 - B. double
 - C. string
 - D. bool

Варианты:

char* или std::string

int

double

bool

float

size_t

№ 11 Прочитайте текст и установите соответствие
Сопоставьте конструкции C# с их функциональностью.

Конструкция

A. if

B. foreach

C. try-catch

D. switch

E. return

Варианты:

Возврат значения из метода

Обработка исключений

Перебор элементов коллекции

Альтернатива множественным if

Условный оператор

Асинхронная операция

№ 12 Прочитайте текст и установите последовательность

Установите порядок операций при использовании массива в C++:

Варианты:

Объявление массива

Заполнение значениями

Использование в цикле

Освобождение памяти (если динамический)

ПК-93 - Способен генерировать новые идеи для решения задач цифровой экономики, абстрагироваться от стандартных моделей, перестраивать сложившиеся способы решения задач, выдвигать альтернативные варианты действий с целью выработки новых оптимальных алгоритмов

- № 1 Прочитайте текст и запишите развернутый обоснованный ответ
Какой конструктор вызывается при создании каждого объекта в массиве, создаваемом оператором `new[]` в C++?
- № 2 Прочитайте текст и запишите развернутый обоснованный ответ
Какой конструктор может быть вызван при создании объекта оператором `new` в C++ ?
- № 3 Прочитайте текст и установите соответствие
Сопоставьте этапы компиляции программы на C++ с их назначением.

Этап

- A. Препроцессинг
- B. Компиляция
- C. Линковка
- D. Исполнение

Варианты:

Объединение объектных файлов и библиотек

Подстановка макросов и включение заголовков

Выполнение машинного кода

Преобразование исходного кода в объектный

Запись данных в файл

Установка точек останова в отладчике

- № 4 Прочитайте текст и установите соответствие
Сопоставьте компоненты .NET-программы на C# с их описанием.

Компонент

- A. CLR (Common Language Runtime)
- B. CTS (Common Type System)
- C. IL (Intermediate Language)
- D. JIT-компилятор
- E. Metadata

Варианты:

Стандартизует типы данных между языками .NET

Хранит информацию о типах, классах, методах

Выполняет IL-код на платформе во время исполнения

Виртуальная машина .NET, управляющая выполнением

Язык, в который компилируется C# перед исполнением

Язык запроса к базе данных

Язык шаблонов ASP.NET

№ 5 Прочитайте текст и установите последовательность

Установите правильную последовательность шагов при вызове метода в C#:

Варианты:

Определение метода

Вызов метода

Передача параметров

Возврат результата

№ 6 Прочитайте текст и установите последовательность

Расположите шаги создания класса и объекта в C#:

Варианты:

Определение класса

Объявление объекта

Инициализация конструктора

Использование методов объекта

№ 7 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

В чём главное отличие объектов классов `System.String` и `System.StringBuilder` в C#?

`System.String` иммутабелен `System.StringBuilder` мутабелен

`System.StringBuilder` иммутабелен `System.String` мутабелен

`System.String` ковариантен `System.StringBuilder` контрвариантен

`System.String` потокобезопасен `System.StringBuilder` нет

№ 8 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Какой из следующих способов организации строки в C# позволит использовать встроенные методы в случае необходимости поиска подстроки?

Используя `System.String`

Используя `System.StringBuilder`

Используя массив `char`

Используя указатель на `char`

№ 9 Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа

Объект какого из перечисленных классов используется для чтения информации из текстового потока в C#?

`StreamReader`

`TextReader`

`TextStream`

`FileReader`

№ 10 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

к ассоциативным контейнерам в C++ относятся

`set`

`map`

`multiset`

`multimap`

`vector`

`list`

`deque`

№ 11 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

к последовательным контейнерам в C++ относятся

vector

list

deque

set

map

string

№ 12 Прочитайте текст, выберите правильные ответы и запишите аргументы, обосновывающие выбор ответов

Какие есть виды итераторов в C++?

входные

выходные

однонаправленные

двунаправленные

произвольного доступа

многопоточные

двумерные

статические

константные