

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Искаков Ирлан Жангазыевич

Автономная некоммерческая организация высшего образования

Должность: Ректор

«Университет при Межпарламентской Ассамблее ЕвразЭС»

Дата подписания: 22.11.2023 10:20:32

Уникальный программный ключ:

a748d5b672796bd7b37612bb23a3449357804892a0d120774ea9def3ef7a2bc0

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

### Программирование на языках высокого уровня

*(наименование дисциплины)*

#### Направление подготовки

09.03.04 Программная инженерия

#### Квалификация выпускника

Бакалавр

#### Направленность (профиль)

Проектирование программного обеспечения

2023 г.

## 1. Планируемые результаты обучения по дисциплине, соотнесенные с индикаторами достижения компетенций

В процессе изучения дисциплины у обучающихся должны быть сформированы следующие компетенции:

**ОПК-1 - способен применять естественно-научные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности**

Индикаторы достижения компетенций	Планируемые результаты обучения
ИД-1 (ОПК-1) применяет языки высокого уровня для профессиональной деятельности	<i>знает</i>
	РО-1 ИД-1 (ОПК-1) теоретические положения объектно-ориентированного программирования ;
	<i>умеет</i>
	РО-2 ИД-1 (ОПК-1) применять языки высокого уровня для решения профессиональных задач
	<i>владеет</i>
	РО-3 ИД-1 (ОПК-1) библиотеками и методами объектно-ориентированных языков для создания программного обеспечения и интерфейсов

## 2. Объем дисциплины в зачетных единицах

Объем дисциплины составляет 7 зачетных единиц.

## 3. Содержание дисциплины

**Раздел 1. Основные понятия объектно-ориентированного программирования (ООП).**1.1. Парадигма ООП: инкапсуляция, наследование, полиморфизм. 1.2. Язык программирования (ЯП) С++ как язык, поддерживающий ООП. 1.3. Сравнение ЯП С++ и С. Указатели и ссылки. Возможность использования фрагментов текстов программ на С в программах на С++. 1.4. Обзор стандартной библиотеки С++. Пространства имен как механизм высокоуровневой инкапсуляции. Стандартные пространства имен. Заголовочные файлы стандартной библиотеки. 1.5. Поточный ввод-вывод в С++. 1.6. Размещение динамических переменных. Операторы new, new[], delete, delete[]. 1.7. Многомерные динамические массивы. 1.8. Операции преобразование типа в С и С++. Операции static\_cast, dynamic\_cast, const\_cast, reinterpret\_cast.

**Раздел 2. Стандартные и пользовательские типы данных в С++. Обработка исключений. Инкапсуляция и статический полиморфизм в С++.** 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.

**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 классы.

**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.

**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.

**Раздел 6. Кроссплатформенная библиотека для создания многооконных приложений SDL 2.x.** 6.1. Модель множества окон отображения в 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.

**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.

**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.

**Раздел 9. Массивы, строки, работа с файлами.** 9.1. Массивы. Определение и инициализация. Допустимые приведения типов массивов. 9.2. Классы System.String и 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.

**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 интерфейс - определение и реализация. Явная и неявная реализация интерфейса. Реализация интерфейсов и наследование.

**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++.

**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.

**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.

**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).

**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.

**Раздел 16. Разработка оконных приложений на языке C#.** 16.1 Технология Windows Forms. Структура проекта. Ресурсы проекта. Архитектурный шаблон MVC. Особенности создания приложения.

#### **4. Методические рекомендации по организации изучения учебной дисциплины, включая самостоятельную работу обучающихся**

Изучение дисциплины включает контактную работу обучающихся с педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях в форме занятий различных типов в соответствии со спецификой дисциплины и самостоятельную работу обучающихся в объемах соответственно учебному плану. Контактная работа может проводиться с применением электронного обучения, дистанционных образовательных технологий.

#### **Вопросы для самоконтроля/Задания для самоконтроля/Вопросы и задания для самоконтроля**

1. Стандартные и пользовательские типы данных в C++. Классы в C++: определение и использование.
2. Наследование. Иерархия классов. Виртуальные функции. Динамический полиморфизм в C++. Виртуальные деструкторы. Оператор typeid.
3. Чисто виртуальные функции и абстрактные базовые классы. Множественное наследование. Ромбовидное наследование. Виртуальные базовые классы.
4. Шаблоны функций и шаблоны классов в C++. Оператор typename.
5. Обобщенное программирование с использованием библиотеки стандартных шаблонов (Standard Template Library, STL) C++
6. Способы программного построения двумерных движущихся изображений
7. Система типов C#, классы: определение и использование
8. Работа с массивами, строками. Определение потоковых классов
9. Сравнение написания классов в C++ и C#. Определение интерфейса.
10. Использование интерфейса в механизме наследования

11. Работа с механизмом исключений. Жизненный цикл объекта.
12. Работа со сборщиком мусора. Поколения жизни объекта
13. Понятие коллекции. Работа с коллекцией. Работа с итераторами.
14. Универсальные интерфейсы коллекций.
15. Механизм делегатов. Реализация событий с использованием делегатов.

**5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

**Основная:**

1. В. И. Юров. . Assembler. СПб.: Питер, 2010, эл. рес.
2. Дж. Бишоп, Н. Хорспул. . C# в кратком изложении. М.: БИНОМ. Лаборатория знаний, 2005, эл.ресурс
3. Н. А. Тюкачев, В. Г. Хлебостроев. . C#. Программирование 2D и 3D векторной графики. Санкт-Петербург: Лань, 2022, эл. рес.
4. С. С. Сосинская. . Использование языка C# в различных информационных технологиях. Старый Оскол: ТНТ, 2020, эл. рес.
5. Т. А. Павловская. . C/C++. Программирование на языке высокого уровня . Санкт-Петербург: Питер, 2021, эл. рес.

**Дополнительная:**

Не требуется

**6. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", информационных справочных систем и профессиональных баз данных, необходимых для освоения дисциплины**

1. <http://biblioclub.ru/> – электронная библиотечная система «Университетская библиотека Онлайн»
2. <http://www.iprbookshop.ru/> – электронная библиотечная система IPR BOOKS
3. <http://www.yurist.ru>
4. <http://www.garant.ru> – ГАРАНТ: [Информационно-правовой портал]
5. Справочная правовая система Консультант Бизнес: Версия Проф  
Профессиональные базы данных в составе СПС Консультант:  
- Законодательство Санкт-Петербурга и Ленинградской области  
- Международное право

**7. Лицензионное программное обеспечение**

- Офисный пакет Libre Office;
- Интернет-браузер Mozilla Firefox;
- Dr.Web Desktop Security Suite (Комплексная защита)
- Moodle 3.8.2.
- Интегрированная среда разработки Code::Blocks

**8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

В зависимости от вида проводимых учебных занятий и форм осуществления образовательной деятельности по соответствующей образовательной программе используется следующее материально-техническое обеспечение дисциплины:

- учебные аудитории для проведения занятий лекционного типа (укомплектованные специализированной мебелью и оборудованные техническими средствами обучения, служащими для представления учебной информации большой аудитории, а также имеющие наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающих тематические иллюстрации, соответствующие рабочим программам дисциплин);

- учебные аудитории для проведения занятий семинарского типа (с типовым оборудованием, обеспечивающим применение современных информационных технологий, и наглядными пособиями);

- специальные помещения для проведения занятий по дисциплине (в т.ч. лаборатории, оснащенные лабораторным оборудованием, в зависимости от степени сложности), а именно;

- компьютерные классы с демонстрационно-обучающими и обучающе-контролирующими возможностями, доступом к базам данных и Интернет;

- кабинет для занятий по иностранному языку (оснащенный лингафонным оборудованием);

- учебные аудитории для групповых и индивидуальных консультаций;

- учебные аудитории для текущего контроля и промежуточной аттестации;

- помещения для самостоятельной работы обучающихся (оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду организации);

- библиотека (имеющая читальные залы и рабочие места для обучающихся, оснащенные компьютерами с доступом к базам данных и Интернет).

Для инвалидов и лиц с ограниченными возможностями здоровья форма проведения занятий по дисциплине устанавливается образовательной организацией с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья. При определении формы проведения занятий с обучающимся-инвалидом образовательная организация должна учитывать рекомендации, данные по результатам медико-социальной экспертизы, содержащиеся в индивидуальной программе реабилитации инвалида, относительно рекомендованных условий и видов труда. При необходимости для обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья создаются специальные рабочие места с учетом нарушенных функций и ограничений жизнедеятельности. При необходимости обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечиваются печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.