

**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное автономное образовательное  
учреждение высшего образования "Пермский  
государственный национальный исследовательский  
университет"**

**Кафедра информационных технологий**

Авторы-составители: **Шестаков Александр Петрович**  
**Залогова Любовь Алексеевна**  
**Кнутова Наталия Сергеевна**  
**Соловьева Татьяна Николаевна**

Рабочая программа дисциплины  
**ЯЗЫКИ ПРОГРАММИРОВАНИЯ**  
Код УМК 68708

Утверждено  
Протокол №6  
от «09» июня 2021 г.

Пермь, 2021

## **1. Наименование дисциплины**

Языки программирования

## **2. Место дисциплины в структуре образовательной программы**

Дисциплина входит в обязательную часть Блока « Б.1 » образовательной программы по направлениям подготовки (специальностям):

Направление: **01.03.02** Прикладная математика и информатика

направленность Математическое моделирование и информационные технологии в экономике

### **3. Планируемые результаты обучения по дисциплине**

В результате освоения дисциплины **Языки программирования** у обучающегося должны быть сформированы следующие компетенции:

**01.03.02** Прикладная математика и информатика (направленность : Математическое моделирование и информационные технологии в экономике)

**ОПК.2** Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

#### **Индикаторы**

**ОПК.2.1** Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности

**ОПК.2.2** Анализирует типовые языки программирования, составляет программы

**ОПК.2.3** Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения

#### 4. Объем и содержание дисциплины

<b>Направления подготовки</b>	01.03.02 Прикладная математика и информатика (направленность: Математическое моделирование и информационные технологии в экономике)
<b>форма обучения</b>	очная
<b>№№ триместров, выделенных для изучения дисциплины</b>	3,4
<b>Объем дисциплины (з.е.)</b>	5
<b>Объем дисциплины (ак.час.)</b>	180
<b>Контактная работа с преподавателем (ак.час.), в том числе:</b>	70
<b>Проведение лекционных занятий</b>	28
<b>Проведение практических занятий, семинаров</b>	14
<b>Проведение лабораторных работ, занятий по иностранному языку</b>	28
<b>Самостоятельная работа (ак.час.)</b>	110
<b>Формы текущего контроля</b>	Защищаемое контрольное мероприятие (4) Итоговое контрольное мероприятие (2)
<b>Формы промежуточной аттестации</b>	Зачет (3 триместр) Экзамен (4 триместр)

## 5. Аннотированное описание содержания разделов и тем дисциплины

### Языки программирования

Определенный способ мышления (парадигма) служит основой для создания языка программирования. Существуют разные парадигмы программирования – процедурная, объектно-ориентированная, функциональная, логическая.

Каждая из парадигм используется для решения определенного класса задач. Некоторые языки поддерживают несколько

парадигм, другие же, наоборот, ориентированы на реализацию только одной парадигмы.

Курс посвящен изучению различных парадигм программирования, а также методам программирования в рамках этих парадигм.

### Функциональное программирование

В последнее время наблюдается рост интереса к функциональному программированию.

Эта парадигма применяется для создания систем искусственного интеллекта, разработки графических интерфейсов,

создания систем параллельных вычислений, а также построения компиляторов.

Функциональная парадигма не должна рассматриваться как замена другим стилям программирования. Она лишь представляет другой подход к разработке программ. Для решения некоторого класса задач такой подход является более эффективным.

В разделе

- изучаются особенности функциональной парадигмы ( неизменность данных, отсутствие побочных эффектов,

использование чистых функций, рекурсия, хвостовая рекурсия, функции высшего порядка и др.)

- применение этих особенностей для реализации приложений на языке функционального программирования.

### Логическое программирование (ПРОЛОГ)

Идея логического стиля программирования заключается в том, чтобы

1. описать совокупность утверждений на формальном языке;
2. воспользоваться системой логического вывода для получения решения.

На языке логического программирования достаточно описать предметную область и поставить цель, а система автоматически найдёт решение (если оно существует). При использовании такого языка основное внимание уделяется описанию объектов и связей между ними, а не разработке последовательности действий для достижения цели.

Программист сообщает системе, что известно и задаёт вопросы. Его в большей степени интересуют знания

и в меньшей – алгоритмы, при помощи которых из этих знаний извлекается информация.

Область применения языков логического программирования- искусственный интеллект.

В разделе изучаются особенности программирования в рамках логической парадигмы.

### Основы компиляции. Итоговое занятие

Компиляторы являются важнейшей областью исследований, связанных с программным обеспечением. Без компиляторов программистам пришлось бы писать программы на автокоде или в машинных кодах.

Цель раздела " Основы компиляции" заключается в том, чтобы

- рассмотреть структуру компилятора, а именно, представить компилятор как совокупность логически взаимосвязанных модулей;
- определить взаимодействие между этими модулями и изучить принципы их построения.

Итоговое занятие посвящено проверке знаний по всем темам курса.

Время выполнения задания - 1 час.

Дистанционная форма сдачи: ответ пишется "от руки",  
фотографируется и  
отправляется по электронной почте преподавателю  
(время - 5 минут).

**Объектно-ориентированная парадигма на примере языка C# (принципы ООП; консольные приложения; массивы; описание классов; создание объектов; передача параметров; наследование)**

В разделе рассматриваются основные принципы объектно-ориентированного программирования - инкапсуляция, наследование, полиморфизм.

Используются консольные приложения, которые наилучшим образом подходят для изучения языка, так как в них нет множества стандартных объектов, необходимых для создания графического интерфейса.

Изучается

- структура классов - шаблонов, на основе которых строятся объекты;
- создание и удаление объектов;
- массивы объектов;
- коллекции объектов.

В разделе рассматривается описание и использование наследования, а также его особенности и достоинства.

Наследование позволяет создать общий класс, который определяет элементы, характерные множеству других классов.

Таким образом, новые классы можно создавать на основе существующего класса-предка. Это, в свою очередь,

позволяет избежать дублирования кода и облегчить редактирование программ.

Изучение теоретического материала сопровождается лабораторными работами, на которых студенты осваивают приемы объектно-ориентированного программирования на базе языка C#.

**Объектно-ориентированная парадигма (файлы, многоуровневые иерархии, полиморфизм)**

Рассматривается построение многоуровневых иерархий. Многоуровневое наследование имеет место, когда производный класс наследует базовому классу, а затем сам становится базовым.

В этом случае иерархия содержит несколько уровней.

Во всех ранее рассмотренных примерах использовались средства консольного ввода/вывода – статические методы ReadLine и WriteLine класса Console пространства имен System.

Однако данные, введенные с клавиатуры и отображенные на экране, доступны лишь во время выполнения программы;

по завершении работы программы все данные теряются. Содержимое же файлов можно использовать многократно.

Кроме того, использование файлов позволяет работать с большими объемами данных.

В разделе рассматривается работа с текстовыми файлами, а также сохранение состояния объектов в бинарном файле.

В иерархии классов может существовать несколько версий одного метода. Это означает, что метод сначала определяется в базовом классе, а затем переопределяется в производных классах.

Интерес представляют ситуации, когда любая из версий переопределенного метода вызывается посредством ссылки на объект базового класса; а решение о том, какую версию метода выполнить принимается динамически – во время выполнения программы. В этом и заключается принцип полиморфизма в ООП.

В разделе рассматривается реализация полиморфизма в рамках иерархии классов.

Изучение теоретического материала сопровождается лабораторными работами, на которых студенты осваивают приемы объектно-ориентированного программирования на базе языка C#.

### **Создание Windows-приложений. Итоговое занятие**

Раздел посвящен разработке объектно-ориентированных приложений с графическим интерфейсом.

Изучаются основные этапы создания таких приложений:

- визуальное проектирование (задание внешнего вида приложения) и
- определение поведения приложения (написание обработчиков событий).

На конкретных примерах демонстрируется использование собственных классов, наследования и полиморфизма для создания приложений с графическим интерфейсом.

## **6. Методические указания для обучающихся по освоению дисциплины**

Освоение дисциплины требует систематического изучения всех тем в той последовательности, в какой они указаны в рабочей программе.

Основными видами учебной работы являются аудиторские занятия. Их цель - расширить базовые знания обучающихся по осваиваемой дисциплине и систему теоретических ориентиров для последующего более глубокого освоения программного материала в ходе самостоятельной работы. Обучающемуся важно помнить, что контактная работа с преподавателем эффективно помогает ему овладеть программным материалом благодаря расстановке необходимых акцентов и удержанию внимания интонационными модуляциями голоса, а также подключением аудио-визуального механизма восприятия информации.

Самостоятельная работа преследует следующие цели:

- закрепление и совершенствование теоретических знаний, полученных на лекционных занятиях;
- формирование навыков подготовки текстовой составляющей информации учебного и научного назначения для размещения в различных информационных системах;
- совершенствование навыков поиска научных публикаций и образовательных ресурсов, размещенных в сети Интернет;
- самоконтроль освоения программного материала.

Обучающемуся необходимо помнить, что результаты самостоятельной работы контролируются преподавателем во время проведения мероприятий текущего контроля и учитываются при промежуточной аттестации.

Обучающимся с ОВЗ и инвалидов предоставляется возможность выбора форм проведения мероприятий текущего контроля, альтернативных формам, предусмотренным рабочей программой дисциплины. Предусматривается возможность увеличения в пределах 1 академического часа времени, отводимого на выполнение контрольных мероприятий.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации.

При проведении текущего контроля применяются оценочные средства, обеспечивающие передачу информации, от обучающегося к преподавателю, с учетом психофизиологических особенностей здоровья обучающихся.

## **7. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине**

При самостоятельной работе обучающимся следует использовать:

- конспекты лекций;
- литературу из перечня основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля);
- текст лекций на электронных носителях;
- ресурсы информационно-телекоммуникационной сети "Интернет", необходимые для освоения дисциплины;
- лицензионное и свободно распространяемое программное обеспечение из перечня информационных технологий, используемых при осуществлении образовательного процесса по дисциплине;
- методические указания для обучающихся по освоению дисциплины.



## 8. Перечень основной и дополнительной учебной литературы

### Основная:

1. Мейер, Б. Основы объектно-ориентированного проектирования : учебник / Б. Мейер. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 751 с. — ISBN 978-54497-0885-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/102030>
2. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие для вузов / Л. А. Залогова. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 192 с. — ISBN 978-5-8114-8481-2. — Текст : электронный // Лань : электронно-библиотечная система. <https://elis.psu.ru/node/538805>
3. Городняя, Л. В. Основы функционального программирования : учебное пособие / Л. В. Городняя. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 246 с. — ISBN 978-5-4497-0932-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/102042>
4. Павловская, Т. А. Программирование на языке высокого уровня C# : учебное пособие / Т. А. Павловская. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 245 с. — ISBN 978-5-4497-0862-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/102051.html>

### Дополнительная:

1. Зыков, С. В. Введение в теорию программирования. Объектно-ориентированный подход : учебное пособие / С. В. Зыков. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 187 с. — ISBN 978-5-4497-0926-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/102007>
2. Братко И. Алгоритмы искусственного интеллекта на языке Prolog/Пер. с англ. и ред. К. А. Птицына.- М.; СПб.; Киев:Вильямс,2004, ISBN 5-8459-0664-4.-640.-Библиогр.: с. 611
3. Шрайнер П. А. Основы программирования на языке Пролог:курс лекций : учеб. пособие/П. А. Шрайнер.-Москва:Интернет-Университет информационных технологий,2005, ISBN 5-9556-0034-5.-176.- Библиогр.: с. 173

## 9. Перечень ресурсов сети Интернет, необходимых для освоения дисциплины

<https://www.intuit.ru/studies/courses/629/485/info> Программирование на языке высокого уровня C#.

<https://www.intuit.ru/studies/courses/471/327/info> Функциональное программирование.

<https://www.intuit.ru/studies/courses/558/414/info> Логическое программирование.

<https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx> Руководство по программированию на C# - MSDN Microsoft

<http://msugvnu000.web710.discountasp.net/Posts/Details/4234> Руководство по изучению F#

<http://www.visual-prolog.com/> Visual Prolog

## 10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Образовательный процесс по дисциплине **Языки программирования** предполагает использование следующего программного обеспечения и информационных справочных систем:

Для формирования и развития профессиональных навыков обучающихся, в учебном процессе используются:

- презентационные материалы (слайды по темам занятий);
- доступ в режиме on-line в Электронную библиотечную систему (ЭБС);
- тестирование;
- доступ в электронную информационно-образовательную среду университета;
- Интернет-сервисы и электронные ресурсы (поисковые системы, электронная почта, профессиональные тематические чаты и форумы, системы аудио и видео конференций, онлайн энциклопедии и т.д.).

Необходимое лицензионное и (или) свободно распространяемое программное обеспечение:

- приложение позволяющее просматривать и воспроизводить медиаконтент PDF-файлов «Adobe Acrobat Reader DC»;
- программа-браузер;
- офисный пакет приложений «LibreOffice», Alt Linux.

Специализированное программное обеспечение: Microsoft Visual Studio, Visual Prolog, SWI-Prolog.

При освоении материала и выполнения заданий по дисциплине рекомендуется использование материалов, размещенных в Личных кабинетах обучающихся ЕТИС ПГНИУ (**student.psu.ru**).

При организации дистанционной работы и проведении занятий в режиме онлайн могут использоваться:

система видеоконференцсвязи на основе платформы BigBlueButton (<https://bigbluebutton.org/>).

система LMS Moodle (<http://e-learn.psu.ru/>), которая поддерживает возможность использования текстовых материалов и презентаций, аудио- и видеоконтент, а так же тесты, проверяемые задания, задания для совместной работы.

система тестирования Indigo (<https://indigotech.ru/>).

## 11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для лекционных занятий требуется аудитория, оснащенная презентационной техникой (проектор, экран, компьютер) с соответствующим программным обеспечением, меловой и/или маркерной доской.

Для проведения лабораторных занятий необходим компьютерный класс, оснащенный соответствующим программным обеспечением, а именно:

Для групповых (индивидуальных) консультаций - аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.

Для проведения текущего контроля - аудитория, оснащенная меловой (и) или маркерной доской.

Для самостоятельной работы студентов необходима: аудитория, оснащенная компьютерной техникой с возможностью подключения к сети «Интернет», с обеспеченным доступом в электронную информационно-образовательную среду университета, помещения Научной библиотеки ПГНИУ.

Помещения научной библиотеки ПГНИУ для обеспечения самостоятельной работы обучающихся:

1. Научно-библиографический отдел, корп.1, ауд. 142. Оборудован 3 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

2. Читальный зал гуманитарной литературы, корп. 2, ауд. 418. Оборудован 7 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

3. Читальный зал естественной литературы, корп.6, ауд. 107а. Оборудован 5 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

4. Отдел иностранной литературы, корп.2 ауд. 207. Оборудован 1 персональным компьютером с доступом к локальной и глобальной компьютерным сетям.

5. Библиотека юридического факультета, корп.9, ауд. 4. Оборудована 11 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

6. Читальный зал географического факультета, корп.8, ауд. 419. Оборудован 6 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

Все компьютеры, установленные в помещениях научной библиотеки, оснащены следующим программным обеспечением:

Операционная система ALT Linux;

Офисный пакет Libreoffice.

Справочно-правовая система «КонсультантПлюс»

**Фонды оценочных средств для аттестации по дисциплине  
Языки программирования**

**Планируемые результаты обучения по дисциплине для формирования компетенции.  
Индикаторы и критерии их оценивания**

**ОПК.2**

**Способен разрабатывать алгоритмы и компьютерные программы, пригодные для  
практического применения**

<b>Компетенция (индикатор)</b>	<b>Планируемые результаты обучения</b>	<b>Критерии оценивания результатов обучения</b>
<b>ОПК.2.2</b> Анализирует типовые языки программирования, составляет программы	умеет анализировать типовые языки программирования, составлять программы	<p style="text-align: center;"><b>Неудовлетворител</b></p> не знает принципы организации типовых языков программирования не умеет составлять алгоритмы и программы <p style="text-align: center;"><b>Удовлетворительн</b></p> разбирается в принципах организации типовых языков программирования владеет навыками анализа поставленной задачи; при составлении алгоритма ее решения допущены ошибки; оформляет алгоритм в виде программы на языке программирования, однако программа составлена нерационально и/или содержит ошибки. <p style="text-align: center;"><b>Хорошо</b></p> знает принципы организации типовых языков программирования; владеет навыками анализа поставленной задачи; составляет алгоритм ее решения; оформляет алгоритм в виде программы на языке программирования, однако программы составляет с ошибками или нерационально. <p style="text-align: center;"><b>Отлично</b></p> знает принципы организации типовых языков программирования; владеет навыками анализа поставленной задачи; составляет алгоритм ее решения; оформляет алгоритм в виде программы на языке программирования.

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
<p><b>ОПК.2.1</b> Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p>	<p>знает и использует в практической деятельности основные положения и концепции в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p>	<p><b>Неудовлетворител</b> Не владеет - знаниями основных положений и концепций в области программирования, - принципами организации языков программирования, - терминологией и базовыми алгоритмами, - средствами информационной безопасности.</p> <p><b>Удовлетворительн</b> - применяет знания основных положений и концепций в области программирования, - не в полной мере владеет принципами организации языков программирования, - допускает ошибки в основной терминологии и базовых алгоритмах, - не соблюдает основные требования информационной безопасности.</p> <p><b>Хорошо</b> - применяет знания основных положений и концепций в области программирования, - владеет принципами организации языков программирования, - допускает ошибки в основной терминологии и базовых алгоритмах, - не полностью соблюдает основные требования информационной безопасности.</p> <p><b>Отлично</b> Применяет знания - основных положений и концепций в области программирования, - архитектуру языков программирования, - основную терминологию и базовые алгоритмы, - основные требования информационной безопасности.</p>
<p><b>ОПК.2.3</b> Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p>	<p>имеет опыт решения задач с использованием базовых алгоритмов, умеет анализировать различные типы программного обеспечения</p>	<p><b>Неудовлетворител</b> не имеет опыта решения задач с использованием базовых алгоритмов; не умеет анализировать различные типы программного обеспечения.</p> <p><b>Удовлетворительн</b> имеет недостаточный опыт решения задач с использованием базовых алгоритмов, допускает грубые ошибки при анализе различных типов программного обеспечения</p>

<b>Компетенция (индикатор)</b>	<b>Планируемые результаты обучения</b>	<b>Критерии оценивания результатов обучения</b>
		<p style="text-align: center;"><b>Хорошо</b></p> <p>имеет опыт решения задач с использованием базовых алгоритмов; допускает ошибки анализа различных типов программного обеспечения</p> <p style="text-align: center;"><b>Отлично</b></p> <p>имеет опыт решения задач с использованием базовых алгоритмов, успешно анализирует различные типы программного обеспечения</p>

## Оценочные средства текущего контроля и промежуточной аттестации

Схема доставки : СУОС +

**Вид мероприятия промежуточной аттестации :** Зачет

**Способ проведения мероприятия промежуточной аттестации :** Оценка по дисциплине в рамках промежуточной аттестации определяется на основе баллов, набранных обучающимся на контрольных мероприятиях, проводимых в течение учебного периода.

**Максимальное количество баллов :** 100

### Конвертация баллов в отметки

«отлично» - от 81 до 100

«хорошо» - от 61 до 80

«удовлетворительно» - от 43 до 60

«неудовлетворительно» / «незачтено» менее 43 балла

Компетенция (индикатор)	Мероприятие текущего контроля	Контролируемые элементы результатов обучения
<b>ОПК.2.1</b> Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности	Функциональное программирование <b>Защищаемое контрольное мероприятие</b>	знание принципов функционального программирования умение составлять программы на функциональном языке F#
<b>ОПК.2.3</b> Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения		
<b>ОПК.2.2</b> Анализирует типовые языки программирования, составляет программы		

<b>Компетенция (индикатор)</b>	<b>Мероприятие текущего контроля</b>	<b>Контролируемые элементы результатов обучения</b>
<p><b>ОПК.2.1</b> Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p> <p><b>ОПК.2.3</b> Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p><b>ОПК.2.2</b> Анализирует типовые языки программирования, составляет программы</p>	<p>Логическое программирование (ПРОЛОГ)</p> <p><b>Защищаемое контрольное мероприятие</b></p>	<p>- знание технологии логического программирования- умение разрабатывать программы на языке логического программирования Prolog</p>
<p><b>ОПК.2.1</b> Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p> <p><b>ОПК.2.3</b> Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p><b>ОПК.2.2</b> Анализирует типовые языки программирования, составляет программы</p>	<p>Основы компиляции.</p> <p>Итоговое занятие</p> <p><b>Итоговое контрольное мероприятие</b></p>	<p>знание основ трансляции программ с языков высокого уровня; владение технологиями функционального программирования; владение технологиями логического программирования;</p>

**Спецификация мероприятий текущего контроля**



## Функциональное программирование

Продолжительность проведения мероприятия промежуточной аттестации: **1 часа**

Условия проведения мероприятия: **в часы аудиторной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **13**

Показатели оценивания	Баллы
создание приложений на F# с использованием концепции вывода типов, условных выражений, рекурсивных функций, кортежей, каррирования, сопоставления с образцом, функций высших порядков.	12
создание приложений на F# для работы со списками и деревьями	10
знание принципов функционального программирования.	8

## Логическое программирование (ПРОЛОГ)

Продолжительность проведения мероприятия промежуточной аттестации: **1 часа**

Условия проведения мероприятия: **в часы аудиторной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **13**

Показатели оценивания	Баллы
создание приложений на языке Prolog, реализующих управление перебором.	12
создание приложений на языке Prolog, реализующих полный перебор.	10
знание основ технологии логического программирования.	8

## Основы компиляции. Итоговое занятие

Продолжительность проведения мероприятия промежуточной аттестации: **2 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **40**

Проходной балл: **17**

Показатели оценивания	Баллы
владение технологиями функционального программирования	15
владение технологиями логического программирования	15
знание основ трансляции программ с языков высокого уровня	10

**Вид мероприятия промежуточной аттестации :** Экзамен

**Способ проведения мероприятия промежуточной аттестации :** Оценка по дисциплине в рамках промежуточной аттестации определяется на основе баллов, набранных обучающимся на контрольных мероприятиях, проводимых в течение учебного периода.

**Максимальное количество баллов :** 100

## Конвертация баллов в отметки

«отлично» - от 81 до 100

«хорошо» - от 61 до 80

«удовлетворительно» - от 43 до 60

«неудовлетворительно» / «незачтено» менее 43 балла

<b>Компетенция (индикатор)</b>	<b>Мероприятие текущего контроля</b>	<b>Контролируемые элементы результатов обучения</b>
<p><b>ОПК.2.1</b> Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p> <p><b>ОПК.2.3</b> Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p><b>ОПК.2.2</b> Анализирует типовые языки программирования, составляет программы</p>	<p>Объектно - ориентированная парадигма на примере языка С# (принципы ООП; консольные приложения; массивы; описание классов; создание объектов; передача параметров; наследование)</p> <p><b>Защищаемое контрольное мероприятие</b></p>	<p>- создание консольных приложений- обработка массивов- описание классов и создание объектов- способы передачи параметров - организация наследования</p>

<b>Компетенция (индикатор)</b>	<b>Мероприятие текущего контроля</b>	<b>Контролируемые элементы результатов обучения</b>
<p><b>ОПК.2.1</b> Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p> <p><b>ОПК.2.3</b> Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p><b>ОПК.2.2</b> Анализирует типовые языки программирования, составляет программы</p>	<p>Объектно - ориентированная парадигма (файлы, многоуровневые иерархии, полиморфизм)</p> <p><b>Защищаемое контрольное мероприятие</b></p>	<p>- умение обрабатывать бинарные и текстовые файлы- умение создавать многоуровневые иерархии- организация полиморфизма</p>
<p><b>ОПК.2.1</b> Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p> <p><b>ОПК.2.3</b> Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p><b>ОПК.2.2</b> Анализирует типовые языки программирования, составляет программы</p>	<p>Создание Windows-приложений. Итоговое занятие</p> <p><b>Итоговое контрольное мероприятие</b></p>	<p>- умение применять технологию ООП для создания windows-приложений;- владение технологией объектно-ориентированного программирования</p>

**Спецификация мероприятий текущего контроля**

**Объектно - ориентированная парадигма на примере языка С# (принципы ООП; консольные приложения; массивы; описание классов; создание объектов; передача параметров; наследование)**

Продолжительность проведения мероприятия промежуточной аттестации: **1 часа**

Условия проведения мероприятия: **в часы аудиторной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **13**

<b>Показатели оценивания</b>	<b>Баллы</b>
умение описывать классы, создавать объекты, работать с объектами; организовывать передачу параметров	9
создание приложений, реализующих принцип наследования	9
знание принципов ООП	6
создание консольных приложений; работа с массивами значений	6

**Объектно - ориентированная парадигма (файлы, многоуровневые иерархии, полиморфизм)**

Продолжительность проведения мероприятия промежуточной аттестации: **2 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **13**

<b>Показатели оценивания</b>	<b>Баллы</b>
создание приложений, реализующих принцип полиморфизма	12
создание приложений, реализующих обработку бинарных и текстовых файлов	10
создание приложений, реализующих многоуровневые иерархии	8

**Создание Windows-приложений. Итоговое занятие**

Продолжительность проведения мероприятия промежуточной аттестации: **2 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **40**

Проходной балл: **17**

<b>Показатели оценивания</b>	<b>Баллы</b>
владение технологией объектно-ориентированного программирования	14
владение технологией функционального программирования	10
владение технологией логического программирования	8
умение применять технологию ООП для создания windows-приложений;	8