

МИНОБРНАУКИ РОССИИ

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

Институт компьютерных наук и технологий

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

Рабочая программа дисциплины
ЯЗЫКИ ПРОГРАММИРОВАНИЯ
Код УМК 95844

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 « С.1 » образовательной программы по направлениям подготовки (специальностям):

Специальность: **10.05.01** Компьютерная безопасность

направленность Разработка защищенного программного обеспечения

3. Планируемые результаты обучения по дисциплине

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

10.05.01 Компьютерная безопасность (направленность : Разработка защищенного программного обеспечения)

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

Индикаторы

ОПК.2.1 Использует знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности для практического применения

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

Специальность	10.05.01 Компьютерная безопасность (направленность: Разработка защищенного программного обеспечения)
форма обучения	очная
№№ триместров, выделенных для изучения дисциплины	3,4
Объем дисциплины (з.е.)	9
Объем дисциплины (ак.час.)	324
Контактная работа с преподавателем (ак.час.), в том числе:	126
Проведение лекционных занятий	70
Проведение лабораторных работ, занятий по иностранному языку	56
Самостоятельная работа (ак.час.)	198
Формы текущего контроля	Защищаемое контрольное мероприятие (4) Итоговое контрольное мероприятие (2)
Формы промежуточной аттестации	Зачет (3 триместр) Экзамен (4 триместр)

5. Аннотированное описание содержания разделов и тем дисциплины

Языки программирования

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

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.

Итоговое контрольное мероприятие

Итоговое занятие посвящено проверке знаний по темам дисциплины.

Объектно-ориентированное программирование. Консольные приложения. Инкапсуляция. Наследование.

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

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.

Объектно-ориентированное программирование. Полиморфизм. Работа с файлами.

В разделе рассматриваются основные принципы объектно-ориентированного программирования - 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.

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

Объектно-ориентированное программирование. Создание Windows-приложений

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.

Основы компиляции. Лексический анализ. Синтаксический анализ.

Компилятор представлен как совокупность логически взаимосвязанных модулей.

В разделе рассматривается

- назначение и структура лексического анализатора;
- взаимодействие лексического анализатора с другими частями компилятора;
- программирование лексического анализатора;
- программирование синтаксического анализатора методом рекурсивного спуска;
- нейтрализация синтаксических ошибок.

Основы компиляции. Семантический анализ. Генерация кода.

В разделе рассматривается

- назначение семантического анализатора
- организация таблиц семантического анализатора
- понятие архитектуры компьютера (с точки зрения разработчика компилятора);
- организация оперативной памяти во время выполнения программы;
- промежуточное представление и генерация кода для выражений;
- промежуточное представление и генерация кода для операторов.

Итоговое контрольное мероприятие

Итоговое занятие посвящено проверке знаний по всем темам дисциплины.

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. Залогова Л. А. Языки программирования. Основы объектно-ориентированного программирования на базе языка C#: учебное пособие для вузов / Л. А. Залогова. - Пермь, 2017, ISBN 978-5-7944-2879-7. - 169.
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. Шрайнер П. А. Основы программирования на языке Пролог: курс лекций: учебное пособие / П. А. Шрайнер. - Москва: Интернет-Университет информационных технологий, 2005, ISBN 5-9556-0034-5. - 176. - Библиогр.: с. 173
4. Ахо Альфред В., Сети, Ульман Джеффри Д. Компиляторы: Принципы, технологии, инструменты: Пер. с англ. / Альфред В. Ахо, Рави Сети, Джеффри Д. Ульман. - М.: Вильямс, 2001, ISBN 5-8459-0189-8. - 768.

Дополнительная:

1. Пратт Теренс, Зелковиц М. Языки программирования. Разработка и реализация / Под ред. А. Матросова. - СПб.: Питер, 2002, ISBN 5-318-00189-0. - 688.
2. Братко И. Алгоритмы искусственного интеллекта на языке Prolog: перевод с английского / И. Братко ; пер. К. А. Птицына. - Москва: Вильямс, 2004, ISBN 5-8459-0664-4. - 640. - Библиогр.: с. 611

9. Перечень ресурсов сети Интернет, необходимых для освоения дисциплины

<http://window.edu.ru/> Единое окно доступа к образовательным ресурсам

<http://www.psu.ru/elektronnye-resursy-dlya-psu> Электронные ресурсы для ПГНИУ

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.).

В процессе изучения дисциплины используются:

- технологии императивного программирования;
- технологии объектно-ориентированного проектирования и программирования;
- технология функционального программирования;
- технология логического программирования;
- технология отладки и тестирования программ;
- технология программирования в среде Microsoft Visual Studio;
- презентационные материалы (слайды по темам лекционных и практических занятий);
- электронная библиотечная система (ЭБС, доступ в режиме on-line);
- электронная информационно-образовательная среда университета.

При освоении материала и выполнении 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:

- Microsoft Visual Studio

- Visual Prolog
- SWI-Prolog.

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 компьютера 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

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
<p>ОПК.2.2 Анализирует типовые языки программирования, составляет программы</p>	<p>знает принципы организации типовых языков программирования, владеет навыками анализа поставленной задачи, умеет оформлять алгоритм в виде программы на языке программирования</p>	<p align="center">Неудовлетворител не умеет - анализировать типовые языки программирования, - выбирать наиболее подходящий язык для решения задачи, - составлять программы</p> <p align="center">Удовлетворительн способен - анализировать типовые языки программирования, - выбирать наиболее подходящий язык для решения задачи, - составлять программы, однако допускает грубые ошибки</p> <p align="center">Хорошо способен - анализировать типовые языки программирования, - выбирать наиболее подходящий язык для решения задачи, - составлять программы, однако допускает некоторые ошибки</p> <p align="center">Отлично успешно - анализирует типовые языки программирования, - выбирает наиболее подходящий язык для решения задачи, - составляет программы,</p>
<p>ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов,</p>	<p>знает базовые алгоритмы, умеет анализировать различные типы программного обеспечения, владеет навыками решения прикладных задач с</p>	<p align="center">Неудовлетворител не имеет опыта решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p>

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
анализа типов коммуникаций и интеграции различных типов программного обеспечения	использованием базовых алгоритмов	<p align="center">Удовлетворительн</p> <p>демонстрирует опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения, однако допускает грубые ошибки</p> <p align="center">Хорошо</p> <p>демонстрирует опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения, однако допускает некоторые ошибки</p> <p align="center">Отлично</p> <p>успешно применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p>
<p>ОПК.2.1 Использует знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности для практического применения</p>	знает основные требования информационной безопасности, владеет принципами организации языков программирования, умеет использовать в практической деятельности основные положения и концепции в области программирования	<p align="center">Неудовлетворител</p> <p>не владеет</p> <ul style="list-style-type: none"> -основными положениями и концепциями в области программирования, -знаниями архитектуры языков программирования, -основной терминологией и базовыми алгоритмами, основными требованиями информационной безопасности <p align="center">Удовлетворительн</p> <p>использует</p> <ul style="list-style-type: none"> -основные положения и концепции в области программирования, -знания архитектуры языков программирования, -основную терминологию и базовые алгоритмы, основные требования информационной безопасности, однако допускает грубые ошибки <p align="center">Хорошо</p> <p>использует</p> <ul style="list-style-type: none"> -основные положения и концепции в области программирования, -знания архитектуры языков программирования,

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
		<p style="text-align: center;">Хорошо</p> <p>-основную терминологию и базовые алгоритмы, основные требования информационной безопасности, однако допускает некоторые ошибки</p> <p style="text-align: center;">Отлично</p> <p>успешно владеет</p> <ul style="list-style-type: none"> -основными положениями и концепциями в области программирования, -знаниями архитектуры языков программирования, -основной терминологией и базовыми алгоритмами, основными требованиями информационной безопасности

Оценочные средства текущего контроля и промежуточной аттестации

Схема доставки : Базовая

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
<p>ОПК.2.2 Анализирует типовые языки программирования, составляет программы</p>	<p>Функциональное программирование Защищаемое контрольное мероприятие</p>	<p>- создание приложений на F# с использованием концепции вывода типов, условных выражений, рекурсивных функций, кортежей, каррирования, сопоставления с образцом, функций высших порядков. - создание приложений на F# для работы со списками и деревьями - знание принципов функционального программирования.</p>
<p>ОПК.2.2 Анализирует типовые языки программирования, составляет программы</p>	<p>Логическое программирование Защищаемое контрольное мероприятие</p>	<p>- знание технологии логического программирования- умение разрабатывать программы на языке логического программирования Prolog</p>
<p>ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения ОПК.2.2 Анализирует типовые языки программирования, составляет программы</p>	<p>Итоговое контрольное мероприятие Итоговое контрольное мероприятие</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: **2 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

Продолжительность проведения мероприятия промежуточной аттестации: **2 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

Итоговое контрольное мероприятие

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

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

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

Проходной балл: **17**

Показатели оценивания	Баллы
владение технологиями логического программирования	15

владение технологиями функционального 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

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

Компетенция (индикатор)	Мероприятие текущего контроля	Контролируемые элементы результатов обучения
<p>ОПК.2.1 Использует знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности для практического применения</p> <p>ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p>ОПК.2.2 Анализирует типовые языки программирования, составляет программы</p>	<p>Основы компиляции. Семантический анализ. Генерация кода. Защищаемое контрольное мероприятие</p>	<p>Программирование лексического анализатора Назначение и структура лексического анализатора Правила написания синтаксического анализатора методом рекурсивного спуска без нейтрализации синтаксических ошибок. Принципы генерации кода Правила написания синтаксического анализатора методом рекурсивного спуска с нейтрализацией синтаксических ошибок. Программирование синтаксического анализатора методом рекурсивного спуска с нейтрализацией синтаксических ошибок Организация таблиц семантического анализатора Программирование синтаксического анализа с нейтрализацией синтаксических ошибок и семантического анализа для различных конструкций языка программирования</p>
<p>ОПК.2.1 Использует знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности для практического применения</p> <p>ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p>ОПК.2.2 Анализирует типовые языки программирования, составляет программы</p>	<p>Итоговое контрольное мероприятие Итоговое контрольное мероприятие</p>	<p>- владение технологией объектно-ориентированного программирования- назначение, структура и программирование различных блоков компилятора</p>

Спецификация мероприятий текущего контроля

Объектно - ориентированное программирование. Создание Windows-приложений

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

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

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

Проходной балл: **13**

Показатели оценивания	Баллы
создание приложений, реализующих принцип наследования умение описывать классы, создавать объекты, работать с объектами; организовывать передачу параметров знание принципов ООП создание консольных приложений; работа с массивами значений	10
владение технологией объектно-ориентированного программирования умение применять технологию ООП для создания windows-приложений;	10
создание приложений, реализующих принцип полиморфизма создание приложений, реализующих обработку бинарных и текстовых файлов создание приложений, реализующих многоуровневые иерархии	10

Основы компиляции. Семантический анализ. Генерация кода.

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

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

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

Проходной балл: **13**

Показатели оценивания	Баллы
Правила написания синтаксического анализатора методом рекурсивного спуска без нейтрализации синтаксических ошибок. Правила написания синтаксического анализатора методом рекурсивного спуска с нейтрализацией синтаксических ошибок. Программирование синтаксического анализатора методом рекурсивного спуска с нейтрализацией синтаксических ошибок	12
Организация таблиц семантического анализатора Программирование синтаксического анализа с нейтрализацией синтаксических ошибок и семантического анализа для различных конструкций языка программирования Принципы генерации кода	12
Программирование лексического анализатора Назначение и структура лексического анализатора	6

Итоговое контрольное мероприятие

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

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

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

Проходной балл: **17**

Показатели оценивания	Баллы
- назначение, структура и программирование различных блоков компилятора	20
- владение технологией объектно-ориентированного программирования	20