

Санкт-Петербургский филиал федерального государственного  
автономного образовательного учреждения высшего образования  
«Национальный исследовательский университет  
"Высшая школа экономики"»

Факультет Санкт-Петербургская школа  
физико-математических и компьютерных наук

**Рабочая программа дисциплины**  
Анализ программ


для образовательной программы «Прикладная математика и информатика»  
направления подготовки 01.03.02 «Прикладная математика и информатика»  
уровень бакалавриат

Разработчик: Ицыксон Владимир Михайлович, [vicysxon@hse.ru](mailto:vicysxon@hse.ru)

Утверждена Академическим руководителем образовательной программы

«31» августа 2018 г.

А.В. Омельченко



---

Санкт-Петербург, 2018

*Настоящая программа не может быть использована другими подразделениями  
университета и другими вузами без разрешения подразделения-разработчика программы*

## 1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает требования к образовательным результатам и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направления 01.03.02 «Прикладная математика и информатика» подготовки бакалавра, обучающихся по бакалаврской программе «Прикладная математика и информатика», изучающих дисциплину «Анализ программ».

Рабочая программа дисциплины разработана в соответствии с:

- Образовательным стандартом НИУ ВШЭ по направлению подготовки 01.03.02 «Прикладная математика и информатика» (уровень бакалавриата), утвержденным ученым советом Национального исследовательского университета «Высшая школа экономики», протокол от 03.03.2017 №02.
- Основной профессиональной образовательной программой «Прикладная математика и информатика» направления подготовки 01.03.02 «Прикладная математика и информатика»;
- Базовым учебным планом университета по образовательной программе «Прикладная математика и информатика», утвержденным в 2018 г.

## 2. Цели освоения дисциплины

Целями освоения программы «Анализ программ» являются:

- подготовка студентов к теоретическому и практическому применению методов анализа программного обеспечения (ПО);
- получение знаний в области статического анализа ПО, моделей кода, систем типов и их применения в анализе ПО;
- Освоение инструментов анализа ПО.

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- Знать основные способы представления программного кода в применении к анализу, основные алгоритмы и методы анализа ПО.
- Уметь решать типичные задачи анализа ПО, как в рамках средств компиляции и трансляции, так и в рамках отдельно стоящих инструментов анализа.
- Иметь навыки (приобрести опыт) разработки и применения анализаторов кода под конкретную прикладную задачу, язык программирования и ограничения.

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ОС НИУ ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
-------------	-------------------	----------------------------------	---	---	--

<p>Способен критически оценивать и переосмысливать накопленный опыт (собственный и чужой), рефлексировать профессиональную и социальную деятельность</p>	УК-9	РБ СД МЦ	<p>Знает основы статического анализа программ Анализирует программы методами различного статического анализа. Использует методы статического и расширенного статического анализа для анализа своих и чужих программ, их эффективности.</p>	<p>Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа</p>	<p>Домашние задания, устный экзамен</p>
<p>Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</p>	ОПК-4	РБ СД МЦ	<p>Знает требования к безопасности программ, соответствующие поставленной задаче. Проводит анализ программного обеспечения на соответствие требованиям безопасности, предъявляемым к программе. Использует методы статического анализа программ для выявления проблем с соответствием требованиям безопасности.</p>	<p>Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа</p>	<p>Домашние задания, устный экзамен</p>
<p>Способен вести письменную и устную коммуникацию на русском и иностранном языках в рамках профессионального и научного общения</p>	ОПК-5	РБ СД МЦ	<p>Знает основные методы статического анализа программ. Умеет вести обсуждение в устном и письменном виде выбора метода анализа программы, подходящего поставленной задаче. Владеет навыками аргументированной дискуссии о выборе оптимального метода анализа программного обеспечения.</p>	<p>Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа</p>	<p>Домашние задания, устный экзамен</p>

Способен собирать, обрабатывать и интерпретировать данные современных научных исследований в области математики и компьютерных наук, необходимых для формирования выводов по соответствующим научным исследованиям	ПК-1	РБ СД МЦ	Знает различные методы представления программного кода. Решает типичные задачи анализа программного обеспечения. Владеет навыками разработки и применения анализаторов кода.	Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа	Домашние задания, устный экзамен
Способен анализировать, писать и редактировать академические и технические тексты на русском и иностранном языках для решения задач профессиональной и научной деятельности в области математики и компьютерных наук	ПК-4	РБ СД МЦ	Знает основные термины и разделы статического анализа программ. Умеет писать технические тексты, содержащие анализ необходимого программного обеспечения, в том числе анализ используемой памяти. Использует и создает технические и академические тексты, посвященные выбору оптимальных методов анализа программ в соответствии с выбранной задачей.	Лекции, подготовка к практическим занятиям, работа на практических занятиях, самостоятельная работа	Домашние задания, устный экзамен

#### 4. Место дисциплины в структуре образовательной программы

Для образовательной программы «Прикладная математика и информатика» направления подготовки 01.03.02 «Прикладная математика и информатика» настоящая дисциплина относится к циклу дисциплин базовой части к разделу дисциплины по выбору.

Для освоения дисциплины необходимы компетенции, полученные в ходе изучения дисциплин:

- Компиляторы;
- Семантики языков программирования;
- Метавычисления.

#### 5. Тематический план учебной дисциплины

Курс рассчитан на 44 часа аудиторной нагрузки, из них 30 часов лекций и 14 часов

практических занятий, общим объемом 3 зачетных единиц (114 часов).

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Основы статического анализа программ	57	15	0	7	35
2	Расширенные вопросы статического анализа программ	57	15	0	7	35
ИТОГО		114	30	0	14	70

## 6. Содержание дисциплины

<u>Раздел 1</u> Основы статического анализа программ	
Тема 1	Введение в предмет курса. Понятия анализа и верификации ПО. Классификация методов анализа ПО. Непротиворечивость, полнота и точность анализа. Проблемы анализа ПО: теорема Тьюринга, теорема Райса. Основные подходы к статическому анализу ПО. Сигнатурный поиск. Язык TIP и его особенности. Понятие абстрактного синтаксического дерева, другие модели ПО.
Тема 2	Анализ ПО на основе типов. Связь классических систем типов и анализа ПО. Системы типов как простейший вид анализа ПО. Полнота и точность анализа на основе систем типов. Понятие анализа на основе ограничений, типичная структура такого анализа. Применение к языку TIP. Способы решения задачи присвоения типов. Задача унификации. Алгоритмы решения задачи унификации. Рекурсивные типы и регулярная унификация. Унификационный солвер для языка TIP.
Тема 3	Монотонный фреймворк. Понятие графа потока управления и моделей на его основе. Понятие чувствительности к потоку управления в анализе ПО. Виды потокочувствительных анализов. Абстрактный домен и абстрактное состояние. Потокочувствительный анализ знаков. Введение в теорию решёток. Понятие частично упорядоченного множества. Верхние и нижние грани. Примеры решёток, применимых в анализе ПО. Анализ знака для TIP на основе теории решёток. Теорема о наименьшей неподвижной точке. Понятие монотонного фреймворка. Алгоритмы поиска неподвижной точки. May- и must- анализ.
Тема 4	Практическое применение потокочувствительных анализов. Анализ живости. Анализ готовых выражений. Анализ занятости. Анализ достижимых значений. Распространение констант. Примеры задач и их решений для языка TIP.

Тема 5	Интервальный анализ. Тривиальная интервальная решётка. Проблемы реализации анализов на основе бесконечных решёток. Widening в применении к решёткам и ограничения, связанные с его применением. Narrowing в применении к решёткам и ограничения, связанные с его применением. Методы объединения widening и narrowing. Модификации алгоритма решения задачи поиска неподвижной точки с применением widening и narrowing.
<u>Раздел 2</u> Расширенные вопросы статического анализа программ	
Тема 1	Анализ ПО, чувствительный к путям исполнения. Понятие чувствительности к пути исполнения. Зависимости по данным и по управлению. Предикаты путей. Интервальный анализ, чувствительный к путям исполнения, проблемы, связанные с ним. Выбор множества предикатов путей в общем случае. Метод уточнения абстракции на основе контрпримеров (CEGAR).
Тема 2	Метод ограниченной проверки моделей. Метод проверки моделей: применение и проблемы. Метод ограниченной проверки моделей как вид статического анализа ПО. Анализ метода ограниченной проверки моделей в сравнении с монотонным фреймворком и CEGAR. Проблема анализа циклов и раскрутка циклов как способ её решения. Задача выполнимости логических формул как способ решения задачи ограниченной проверки моделей. SAT/SMT-решатели, их ограничения и применение. Теории в основе задачи SMT и их влияние на анализ. Статическое однократное присваивание как модель кода, её преимущества в применении к рассмотренным методам анализа. Сложность и разрешимость задачи ограниченной проверки моделей ПО. Интерпретация контрпримеров SMT-решателя, проблемы и решения.
Тема 3	Межпроцедурный анализ ПО. Понятие межпроцедурности в анализе ПО. Пессимистичный межпроцедурный анализ и проблемы, связанные с ним. Полная подстановка тела функции и проблемы, связанные с ней. Понятие полного графа потока управления для программы и анализ на его основе. Понятие контекстной чувствительности. Метод ограниченного клонирования процедур, его ограничения и параметризация. Методы улучшения контекстной чувствительности. Метод контекстной чувствительности на основе входных данных, его анализ в сравнении с методом ограниченного клонирования процедур. Понятие аппроксимации функции в программе. Проблема внешних функций и варианты её решения.
Тема 4	Анализ памяти Проблема анализа кода с использованием памяти и указателей. Анализ указателей: анализ формы, анализ псевдонимов. Анализ динамического выделения памяти, проблемы контекстной чувствительности. Анализ нулевых указателей.
Тема 5	Анализ указателей Расширенное определение понятия анализа указателей. Анализ псевдонимов. Анализ цели. Анализ формы. Направленный и ненаправленный анализ указателей. Алгоритм Стенсгаарда, его реализация на основе унификации. Кубический фреймворк, постановка задачи и алгоритм её решения. Алгоритм Андерсена, его реализация на основе кубического фреймворка. Контекстно-чувствительный анализ указателей. Чувствительный к потоку управления анализ указателей. Анализ указателей на основе типов. Использование анализа указателей.

## 7. Оценочные средства

### 7.1. Формы контроля знаний студентов

Тип контроля	Форма контроля	4 год	Параметры
		3 модуль	
Текущий	Домашнее задание №1	*	Письменное домашнее задание
	Домашнее задание №2	*	Письменное домашнее задание
	Домашнее задание №3	*	Письменное домашнее задание
	Домашнее задание №4	*	Письменное домашнее задание
	Домашнее задание №5	*	Письменное домашнее задание
	Домашнее задание №6	*	Письменное домашнее задание
Итоговый	Устный экзамен	*	Экзамен в устной форме

### 7.2. Критерии и шкалы оценки, примеры заданий

#### 7.2.1. Текущий контроль

Текущий контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств.

##### Домашнее задание №1

*Домашнее задание №1 выдается студентам в одном варианте из 4 заданий. Каждому заданию присвоен свой балл. Форма представления обучающимися домашнего задания - представленные в письменном виде ответы на вопросы и программный код в системе контроля версий.*

##### Пример домашнего задания №1:

Завершить реализацию класса TypeAnalysis в ТПР. Подумайте, как будет работать анализ, если в программе есть рекурсивный тип (4 балла).

Попробуйте добавить в описанную в лекции систему типов тип Bool, не меняя синтаксис языка. Перепишите правила так, чтобы они продолжали работать на основе имеющегося алгоритма унификации (3 балла).

Реализуйте пункт 2 в рамках пункта 1 (3 балла).

Попробуйте добавить в описанную в лекции 2 (модуль 1, тема 2) систему типов тип «массив». Перепишите правила и приведите результат типизации программы из лекции (3 балла).

##### Критерии оценивания и шкала оценки домашнего задания №1

Оценка	Критерии выставления оценки
--------	-----------------------------

«Отлично» (8-10)	Решено заданий на 12 и более баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы аргументированы и не содержат ошибок.
«Хорошо» (6-7)	Решено заданий на 11 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы содержат не более 2 ошибок
«Удовлетворительно» (4-5)	Решено заданий на 10 баллов. Программный код работает в соответствии с предъявляемыми требованиями и содержит не более двух ошибок
«Неудовлетворительно» (0-3)	Решено заданий менее чем на 10 баллов. Программный код не является синтаксически корректным, не решает поставленную задачу или содержит другие ошибки. Ответы на вопросы отсутствуют, недостаточно аргументированы или содержат ошибки.

### Домашнее задание №2

*Домашнее задание №2 выдается студентам в одном варианте из 3 заданий. Каждому заданию присвоен свой балл. Форма представления обучающимися домашнего задания - представленные в письменном виде ответы на вопросы и программный код в системе контроля версий.*

#### Пример домашнего задания №2:

Выразим ли анализ типов в монотонном фреймворке? А наоборот? Приведите примеры. (3 балла)

Закончите реализацию метода `transfer` в трейте `IntraprocSignAnalysisFunctions` в сопроводительных материалах. (4 балла)

Реализуйте класс `PowersetLattice` в файле `GenericLattices.scala` в сопроводительных материалах. (3 балла)

### Критерии оценивания и шкала оценки домашнего задания №2

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Решено заданий на 10 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы аргументированы и не содержат ошибок.
«Хорошо» (6-7)	Решено заданий на 9 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы содержат не более 2 ошибок
«Удовлетворительно» (4-5)	Решено заданий на 8 баллов. Программный код работает в соответствии с предъявляемыми требованиями и содержит не более двух ошибок
«Неудовлетворительно» (0-3)	Решено заданий менее чем на 8 баллов. Программный код не является синтаксически корректным, не решает поставленную задачу или содержит другие ошибки. Ответы на вопросы отсутствуют, недостаточно аргументированы или содержат ошибки.

### Домашнее задание №3

*Домашнее задание №3 выдается студентам в одном варианте из 3 заданий. Каждому заданию присвоен свой балл. Форма представления обучающимися домашнего задания - представленные в письменном виде ответы на вопросы и программный код в системе контроля версий.*

#### Пример домашнего задания №3:



Рассмотрим плоскую решётку над типами {Bool,Int} из задания 1. Если реализовать для неё анализ, каким по предложенной классификации он будет? Опишите правила вывода. (3 балла)

Закончите реализацию анализа живости переменных в файле LiveVarsAnalysis.scala в сопроводительных материалах. (3 балла)

Реализуйте анализ достижимых определений в файле ReachingDefsAnalysis.scala в сопроводительных материалах. (4 балла)

### Критерии оценивания и шкала оценки домашнего задания №3

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Решено заданий на 10 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы аргументированы и не содержат ошибок.
«Хорошо» (6-7)	Решено заданий на 9 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы содержат не более 2 ошибок
«Удовлетворительно» (4-5)	Решено заданий на 8 баллов. Программный код работает в соответствии с предъявляемыми требованиями и содержит не более двух ошибок
«Неудовлетворительно» (0-3)	Решено заданий менее чем на 8 баллов. Программный код не является синтаксически корректным, не решает поставленную задачу или содержит другие ошибки. Ответы на вопросы отсутствуют, недостаточно аргументированы или содержат ошибки.

### Домашнее задание №4

*Домашнее задание №4 выдается студентам в одном варианте из 5 заданий. Каждому заданию присвоен свой балл. Форма представления обучающимися домашнего задания - представленные в письменном виде ответы на вопросы и программный код в системе контроля версий.*

#### Пример домашнего задания №4:

Допустим, мы хотим реализовать оптимизирующий компилятор для языка TIR. Среди прочего, для работы ему требуется информация о размерах различных переменных: bool (1 bit), byte (8 bit signed), char (16 bit unsigned), int (32 bit signed), bigint (any integer), any (any other thing).

Предложите решетку для реализации анализа размера переменных. Обратите внимание, что необходимо описать не только решетку для одного абстрактного значения, но и все другие решетки, требуемые для анализа целой программы (2 балла)

Опишите правила вычисления различных выражений (3 балла)

Придумайте нетривиальный пример программы на TIR для получившегося анализа и посмотрите, что для него получается (2 балла)

Завершите реализацию метода widen в классе IntervalAnalysisWorklistSolverWithWidening в сопроводительных материалах (2 балла)

Реализуйте предложенный вами анализ размера переменных в файле VariableSizeAnalysis.scala в сопроводительных материалах (4 балла)

### Критерии оценивания и шкала оценки домашнего задания №4

Оценка	Критерии выставления оценки
--------	-----------------------------

«Отлично» (8-10)	Решено заданий на 11 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы аргументированы и не содержат ошибок.
«Хорошо» (6-7)	Решено заданий на 10-9 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы содержат не более 2 ошибок
«Удовлетворительно» (4-5)	Решено заданий на 8 баллов. Программный код работает в соответствии с предъявляемыми требованиями и содержит не более двух ошибок
«Неудовлетворительно» (0-3)	Решено заданий менее чем на 8 баллов. Программный код не является синтаксически корректным, не решает поставленную задачу или содержит другие ошибки. Ответы на вопросы отсутствуют, недостаточно аргументированы или содержат ошибки.

### Домашнее задание №5

*Домашнее задание №5 выдается студентам в одном варианте из 2 заданий. Каждому заданию присвоен свой балл. Форма представления обучающимися домашнего задания - представленные в письменном виде ответы на вопросы и программный код в системе контроля версий.*

#### Пример домашнего задания №5:

Напишите вариант программы, для которой анализ открытости-закрытости файлов (приведенный в материалах к лекции 6 (модуль 2, тема 1)) не показывает корректный результат даже с учётом всех возможных условий в переходах (5 баллов)

Предложите, каким образом можно решить описанные в лекции проблемы в этой ситуации (5 баллов)

### Критерии оценивания и шкала оценки домашнего задания №5

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Решено заданий на 10 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы аргументированы и не содержат ошибок.
«Хорошо» (6-7)	Решено заданий на 9 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы содержат не более 2 ошибок
«Удовлетворительно» (4-5)	Решено заданий на 8 баллов. Программный код работает в соответствии с предъявляемыми требованиями и содержит не более двух ошибок
«Неудовлетворительно» (0-3)	Решено заданий менее чем на 8 баллов. Программный код не является синтаксически корректным, не решает поставленную задачу или содержит другие ошибки. Ответы на вопросы отсутствуют, недостаточно аргументированы или содержат ошибки.

### Домашнее задание №6

*Домашнее задание №6 выдается студентам в одном варианте из 2 заданий. Каждому заданию присвоен свой балл. Форма представления обучающимися домашнего задания - представленные в письменном виде ответы на вопросы и программный код в системе контроля версий.*

#### Пример домашнего задания №6:

Напишите вариант программы, для которой контекстно-чувствительный анализ

знаков требует коэффициент  $k > 1$  (5 баллов)

Приведите пример решётки, для которой контекстно-чувствительный анализ в функциональном стиле является более ресурсозатратным, чем контекстно-чувствительный анализ по месту вызова с глубиной 2 (5 баллов)

### Критерии оценивания и шкала оценки домашнего задания №6

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Решено заданий на 10 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы аргументированы и не содержат ошибок.
«Хорошо» (6-7)	Решено заданий на 9 баллов. Программный код является синтаксически корректным и работает в соответствии с предъявляемыми требованиями. Ответы на вопросы содержат не более 2 ошибок.
«Удовлетворительно» (4-5)	Решено заданий на 8 баллов. Программный код работает в соответствии с предъявляемыми требованиями и содержит не более двух ошибок.
«Неудовлетворительно» (0-3)	Решено заданий менее чем на 8 баллов. Программный код не является синтаксически корректным, не решает поставленную задачу или содержит другие ошибки. Ответы на вопросы отсутствуют, недостаточно аргументированы или содержат ошибки.

### 7.2.2. Итоговый контроль по дисциплине

Проверка качества освоения дисциплины производится в форме устного экзамена.

#### УСТНЫЙ ЭКЗАМЕН

Устный экзамен проводится в форме ответов на вопросы экзаменационного билета. На подготовку ответа выделяется 1 час.

#### Примерный перечень вопросов к экзамену:

1. Понятия анализа и верификации ПО. Классификация методов анализа ПО. Непротиворечивость, полнота и точность анализа. Проблемы анализа ПО: теорема Тьюринга, теорема Райса.
2. Основные подходы к статическому анализу ПО. Сигнатурный поиск. Понятие абстрактного синтаксического дерева, другие модели ПО.
3. Анализ ПО на основе типов. Связь классических систем типов и анализа ПО. Системы типов как простейший вид анализа ПО. Полнота и точность анализа на основе систем типов. Понятие анализа на основе ограничений, типичная структура такого анализа.
4. Способы решения задачи присвоения типов. Задача унификации. Алгоритмы решения задачи унификации. Рекурсивные типы и регулярная унификация.
5. Понятие графа потока управления и моделей на его основе. Понятие чувствительности к потоку управления в анализе ПО. Виды потокочувствительных анализов. Абстрактный домен и абстрактное состояние. Потокочувствительный анализ знаков.
6. Введение в теорию решёток. Понятие частично упорядоченного множества. Верхние и нижние грани. Примеры решёток, применимых в анализе ПО. Анализ знака на основе теории решёток.
7. Теорема о наименьшей неподвижной точке. Понятие монотонного фреймворка. Алгоритмы поиска неподвижной точки. May- и must- анализ.
8. Анализ живости. Анализ готовых выражений.
9. Анализ занятости. Анализ достижимых значений. Распространение констант.
10. Интервальный анализ. Тривиальная интервальная решётка. Проблемы

- реализации анализов на основе бесконечных решёток. Widening в применении к решёткам и ограничения, связанные с его применением.
11. Narrowing в применении к решёткам и ограничения, связанные с его применением. Методы объединения widening и narrowing. Модификации алгоритма решения задачи поиска неподвижной точки с применением widening и narrowing.
  12. Понятие чувствительности к пути исполнения. Зависимости по данным и по управлению. Предикаты путей. Интервальный анализ, чувствительный к путям исполнения, проблемы, связанные с ним. Выбор множества предикатов путей в общем случае. Метод уточнения абстракции на основе контрпримеров (CEGAR).
  13. Метод ограниченной проверки моделей как вид статического анализа ПО. Метод ограниченной проверки моделей в сравнении с монотонным фреймворком и CEGAR.
  14. Метод ограниченной проверки моделей. Проблема анализа циклов и раскрутка циклов как способ её решения. Задача выполнимости логических формул как способ решения задачи ограниченной проверки моделей.
  15. SAT/SMT-решатели, их ограничения и применение. Теории в основе задачи SMT и их влияние на анализ. Интерпретация контрпримеров SMT-решателя, проблемы и решения.
  16. Статическое однократное присваивание как модель кода, её преимущества в применении к рассмотренным методам анализа. Сложность и разрешимость задачи ограниченной проверки моделей ПО.
  17. Понятие межпроцедурности в анализе ПО. Пессимистичный межпроцедурный анализ и проблемы, связанные с ним. Полная подстановка тела функции и проблемы, связанные с ней. Понятие полного графа потока управления для программы и анализ на его основе.
  18. Понятие контекстной чувствительности. Метод ограниченного клонирования процедур, его ограничения и параметризация. Методы улучшения контекстной чувствительности.
  19. Метод контекстной чувствительности на основе входных данных, его анализ в сравнении с методом ограниченного клонирования процедур. Понятие аппроксимации функции в программе. Проблема внешних функций и варианты её решения.
  20. Анализ кода с использованием памяти и указателей. Анализ указателей: анализ формы, анализ псевдонимов. Анализ динамического выделения памяти, проблемы контекстной чувствительности. Анализ нулевых указателей.
  21. Анализ указателей. Анализ псевдонимов. Анализ цели. Анализ формы. Направленный и ненаправленный анализ указателей.
  22. Алгоритм Стенсгаарда, его реализация на основе унификации.
  23. Алгоритм Андерсена, его реализация на основе кубического фреймворка. Контекстно-чувствительный анализ указателей.
  24. Чувствительный к потоку управления анализ указателей. Анализ указателей на основе типов.

### **Критерии оценивания и шкала оценки устного экзамена**

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Дан развернутый ответ на поставленный вопрос. Материал изложен последовательно. Имеются логичные и аргументированные выводы.

«Хорошо» (6-7)	Дан развернутый ответ на поставленный вопрос. Материал изложен непоследовательно. Имеются логичные и аргументированные выводы.
«Удовлетворительно» (4-5)	Ответ на вопрос не является полным. Материал изложен непоследовательно. Отсутствуют выводы.
«Неудовлетворительно» (0-3)	Ответ на вопрос является неверным. Материал изложен непоследовательно. Отсутствуют выводы.

### 7.3 Порядок формирования оценок по дисциплине

Преподаватель учитывает работу на практических занятиях и оценку за текущий контроль (домашние задания).

Результирующая оценка за дисциплину рассчитывается следующим образом:

$$O_{\text{Результирующая}} = 0,45 * O_{\text{накопленная}} + 0,55 * O_{\text{экзамен}}$$

где  $O_{\text{накопленная}}$  рассчитывается как взвешенная сумма всех форм текущего контроля.

$$O_{\text{накопленная}} = \frac{Oд/з1 + Oд/з2 + Oд/з3 + Oд/з4 + Oд/з5 + Oд/з6}{6}$$

Действует следующий способ округления накопленной оценки за текущий контроль: при значениях от 0,1 до 0,4 оценка округляется в меньшую сторону, от 0,5 до 0,9 – в большую.

Результирующая оценка округляется в меньшую сторону.

На экзамене студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

## 8. Образовательные технологии

Основными образовательными технологиями являются: интерактивные лекции, работа в группах на практических занятиях.

## 9. Учебно-методическое и информационное обеспечение дисциплины

### 9.1 Основная литература

1. Software exorcism : a handbook for debugging and optimizing legacy code / Blunden, Bill. Apress. 2012
2. Source Code Analytics With Roslyn and JavaScript Data Visualization / Sudipta Mukherjee; DeWolf. Apress. 2016

### 9.2 Дополнительная литература

1. Write Great Code: Understanding the Machine; Volume I / Randall Hyde. No Starch Press. 2004
2. Testing code security / By: van der Linden, Maura A. CRC Press. 2007
3. From Tracking Code to Analysis: Generalised Courant-Snyder Theory for Any Accelerator Model / Forest, Etienne; Niko. Springer. 2016

## 10. Рекомендации для самостоятельной работы студентов

Самостоятельная работа может рассматриваться как организационная форма обучения – система педагогических условий, обеспечивающих управление учебной деятельностью по освоению знаний и умений в области учебной деятельности без посторонней помощи. Студенту нужно четко понимать, что самостоятельная работа – не

просто обязательное, а необходимое условие для получения знаний по дисциплине и развитию компетенций, необходимых в будущей профессиональной деятельности.

Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных на лекциях теоретических знаний;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- формирования практических (общеучебных и профессиональных) умений и навыков;
- развития исследовательских умений;
- получения навыков эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности.

В учебном процессе выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа - планируемая учебная работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная программой учебной дисциплины, раскрывающей и конкретизирующей ее содержание, осуществляется студентом инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует источники для работы, характеризует наиболее рациональную методику самостоятельной работы, демонстрирует ранее выполненные студентами работы и т. п.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать индивидуальные особенности студента.

Самостоятельная работа может осуществляться индивидуально или группами студентов online и на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности.

Контроль результатов внеаудиторной самостоятельной работы осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине на практических занятиях.

## **11. Материально-техническое обеспечение дисциплины и информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения информационных справочных систем (при необходимости).**

Для проведения всех занятий используется проектор и компьютер для проекции слайдов. Для свободной работы необходимо следующее ПО: Haskell, Ocaml.

## **12. Особенности организации обучения для лиц с ограниченными возможностями здоровья**

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться следующих варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

1) *для лиц с нарушениями зрения:* в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

2) *для лиц с нарушениями слуха:* в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

3) *для лиц с нарушениями опорно-двигательного аппарата:* в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.