**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Основы программирования

Introduction to Programming

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 5

Регистрационный номер рабочей программы: 003574

2021

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Обучение основам программирования и дисциплины программной инженерии.

Задачи курса:

• изучение основных понятий программирования;

• обеспечение базы для усвоения дальнейших курсов по специальности.

Дисциплина «Основы программирования» является базовой в подготовке профессионального программиста, не требуется никакого предварительного опыта в области программирования или информатики. Обучающиеся должны уметь пользоваться математической нотацией и формализмами.

В первом семестре изучаются в соответствующем объёме темы, рекомендуемые Computing Curricula для курса CS101 «Основы программирования». Материал второго семестра основан на рекомендациях по курсам «Объекты и алгоритмы» (CS112F) и «Программная инженерия» (SE101). Первый курс представлен достаточно полно, за счет того, что не рассматриваются темы, входящие в курс «Алгоритмы и структуры данных» (CS103), и сокращенно даются темы, уже изученные в предыдущем семестре. Кроме того, в течение первого и второго семестров закрепление материала происходит в активной форме в рамках курса «Практикум на ЭВМ» и продолжается в течение следующего года.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа курса предназначена для обучающихся 1 курса бакалавриата «Программная инженерия».

Максимальная эффективность программы будет обеспечена при условии, что обучающийся

• знает содержание дисциплины и имеет достаточно полное представление о возможностях применения её разделов в различных прикладных областях науки и техники;

• умеет писать программы согласно представленному алгоритму.

**1.3. Перечень результатов обучения (learning outcomes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование категории (группы) компетенций | Код и наименование компетенции | Планируемые результаты обучения, обеспечивающие формирование компетенции | Код индикатора и индикатор достижения универсальной компетенции |
|  | 1 | 2 | 3 | 4 |
| 1 | Общепрофессиональные компетенции | ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности | знает устройство ЭВМ | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения |
|  | Общепрофессиональные компетенции | ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности | знает основные понятия программирования, | ОПК-2.1 Уметь писать программный код с использованием языков программирования, определения и манипулирования данными |
|  | Общепрофессиональные компетенции | ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов; | знает парадигмы программирования | ОПК-6.1 Уметь проверять и отлаживать программный код |
|  | Профессиональные компетенции | ПКА-1 Способен использовать в педагогической деятельности научные основы образования в сфере ИКТ | знает устройство ЭВМ | ПКА-1.1 Быть способным осуществлять организационное и технологическое обеспечение кодирования на языках программирования |
|  | Профессиональные компетенции | ПКП-1 Способен проектировать программные системы | знает парадигмы программирования | ПКП-1.1 Уметь разрабатывать технических спецификаций на программные компоненты и их взаимодействие |
|  | Профессиональные компетенции | ПКП-2 Способен использовать основные модели информационных технологий и способы их применения для решения задач в предметных областях | знает основные понятия программирования, | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы |
|  | Универсальные компетенции | УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач | знает главные аспекты разработки программного обеспечения | УК 1.4. Оценивает достоинства, недостатки и последствия вариантов решения поставленных задач; |
|  | Универсальные компетенции | УКБ-3 Способен понимать сущность и значение информации в развитии общества, использовать основные методы получения и работы с информацией с учетом современных технологий цифровой экономики и информационной безопасности | знает главные аспекты разработки программного обеспечения | УК-3.3. Строит продуктивное взаимодействие с учетом возможных последствий личных действий в социальном взаимодействии и командной работе; |

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Лекции – 20 ак.ч.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 1 | 30 |  |  |  |  |  |  |  | 2 |  |  |  | 36 |  | 4 |  | 10 | 2 |
|  | 2-45 |  |  |  |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| Семестр 2 | 30 |  | 2 |  |  |  |  |  | 2 |  |  |  | 46 |  | 28 |  | 10 | 3 |
|  | 2-45 |  | 2-25 |  |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 60 |  | 2 |  |  |  |  |  | 4 |  |  |  | 82 |  | 32 |  |  | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 1 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |
| Семестр 2 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

Модуль 1: Базовые понятия программирования (10 ч. лекций),

1. Информация. Синтаксис. Семантика. Прагматика. Количество информации.
2. Данные. Представление данных в памяти ЭВМ: целые (3 способа), представление вещественных (с плавающей точкой). Арифметика с насыщением, вещественные фиксированной точности, двоично-десятичное представление.
3. Транслятор. Интерпретатор. Компилятор. JIT-компилятор. Основные этапы построения исполнимого модуля из исходного текста программы. Роль и принцип работы редактора связей.
4. Операционная система. Ядро. Исполнение кода в современных ОС. Исполнимый модуль ОС и динамические библиотеки. Процессы и нити.
5. Виртуальная память. Полезные свойства для прикладного программиста и принципиальное устройство. Адресное пространство процесса.
6. Файлы, файловые системы. Файловая система FAT: организация, проблемы и способы решения в современных файловых системах.
7. Процедурное программирование. Базовые понятия. Основные управляющие конструкции. Операторы и операции.
8. Процедуры. Параметры. Глобальные и локальные переменные. Организация кадра.

Модуль 2: Архитектура ЭВМ(10 ч. лекций).

1. Архитектура ЭВМ. Принстонская и гарвардская архитектуры. Виртуальные машины, типы виртуализации.
2. Представление значений в памяти: числа, символы, литералы. Адресная арифметика.
3. Составные типы данных: массивы, структуры, объединения. Представление в памяти ЭВМ. Приведение типов.
4. Динамические структуры данных. Методы выделения (и освобождения памяти): «близнецы» и «блоками фиксированного размера»
5. «Мусор» и «трёхцветный» алгоритм сборки. Разновидности сборщиков мусора.
6. Абстрактный тип данных. Интерфейс и реализация. Реализация стека и очереди через (односвязный) список и массив.
7. Реализация индексируемой коллекции на основе массива, списка и развернутого списка. Оценка времени выполнения основных операций при разных реализациях.
8. Основные конструкции управления вычислениями. Циклы с пред- и с пост-условием, цикл for, операторы goto и switch. Графическое изображение (блок-схема, граф потока управления).
9. История ЭВМ.
10. Базы данных.
11. Понятие алгоритма. Машина Тьюринга. Тезис Чёрча-Тьюринга. Формальные определения алгоритма (процесса вычисления, данных).
12. Лямбда-исчисление Чёрча. Стратегии редукции (Call-by-name, Call-by-value, Call-by-reference, Call-by-need). Нумералы Чёрча.
13. Граф потока управления. Структурное программирование. Цикл с пост-условием, цикл с предусловием, ветвление, безусловный переход, вызов подпрограммы, исключенния. Оптимизации на графе потока управления.
14. Процессы и нити. Параллельные вычисления (примеры). Механизмы распараллеливания.

Модуль 3: Разработка программного обеспечения (20 ч. лекций).

1. Программная инженерия. Проблемная область. Определение. Кодекс этики. Отличие от других инженерных специальностей. Кодекс этики АСМ.
2. Жизненный цикл ПО. Модели разработки ПО. Процесс разработки ПО. Роли в проекте.
3. Интеллектуальная собственность. Авторские права. Свободное и бесплатное ПО. Лицензии GNU GPL, BSD(MIT) и Creative Commons.
4. Введение в оценку проектов.
5. Основы делового общения, сетевой этикет.
6. Системы контроля версий. Виды, история развития. Решаемые задачи. Работа с git и GitHub (git workflow, branch per feature).
7. Тестирование и обеспечение качества. Цели, задачи, средства. Ручное и автоматизированное тестирование. Интеграционное тестирование. Модульное тестирование. Методологии TDD и BDD: основные идеи.
8. Постановка экспериментов: повторяемость, воспроизводимость. Достоверность результата.
9. Технологии распределённых реестров (блокчейн, DAG). Принципиальное устройство, решение задачи консенсуса, известные проблемы.
10. Основы компьютерных сетей: Ethernet, IP, UDP, TCP, HTTP, HTTPS(SSL/TLS, PKI), VPN.
11. Основные понятия информационной безопасности. Угрозы. Идентификация и авторизация. Примеры биометрической идентификации. Шифрование с симметричным ключом(AES). Шифрование с открытым ключом (RSA). Цифровая подпись.

Модуль 4: Объектно-ориентированный подход (20 ч. лекций).

1. Парадигмы программирования. История, примеры, языки.
2. Императивная, процедурная и структурная парадигма. Теорема Бёма-Якопини
3. Объектно-ориентированный подход. Основные понятия.
4. Полиморфизм в ООП. Наследование. Раннее и позднее связывание. Статический полиморфизм
5. Объектно-ориентированная парадигма: история развития
6. Инкапсуляция: цели и механизмы.
7. Наследование, агрегация, композиция. Примеры. Множественное наследование: различные подходы в языках программирования
8. Абстрактные классы. Абстрактные и виртуальные методы. Интерфейсы
9. Принципы SOLID объектно-ориентированного проектирования.
10. Модель приложения MVC: компоненты и их назначение

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Методические материалы включают в себя следующие типы материалов — учебники, учебные пособия, методические указания для обучающихся, Интернет-ресурсы, электронные учебные пособия, с опорой на которые проводится аудиторная работа.

Во время занятий могут цитироваться и демонстрироваться выдержки из следующих источников:

1. Куликов Е.К., Макаров А.А. Элементы параллельного программирования (учебное пособие). – СПб: ООО «Политехника-принт», 2018. – 60 с. Мм – 1 экз.

2. Царев, Р. Ю. Алгоритмы и структуры данных (CDIO): Учебник / Царев Р.Ю., Прокопенко А.В. - Краснояр.:СФУ, 2016. - 204 с.: ISBN 978-5-7638-3388-1. - Текст : электронный. - URL: https://znanium.com/catalog/product/967108

**3.1.2 Методическое обеспечение самостоятельной работы**

Самостоятельная работа обучающихся, как вид деятельности, стимулирующий активность, самостоятельность, познавательный интерес с целью поиска необходимой информации, приобретения знаний, использования этих знаний для решения учебных, научных и профессиональных задач, представляет собой важную составляющую учебного процесса, которой отводится не менее половины учебного времени при очной форме обучения. Время, отводимое на самостоятельную работу, должно использоваться обучающимися для наиболее полного освоения учебной дисциплины. Следовательно, организация эффективной внеаудиторной самостоятельной работы в процессе обучения требует, с одной стороны, создание условий, призванных обеспечить рациональное и планомерное управление учебной деятельностью, протекающей в отсутствие преподавателя, и тщательного подбора целого ряда учебных пособий, с другой стороны.

Роль преподавателя в организации самостоятельной работы состоит в координации действий обучающихся в освоении дисциплины, в методическом и организационном обеспечении учебного процесса. Взаимодействие между преподавателем и обучающимися осуществляется в форме консультаций. Преподаватели также оказывают помощь обучающимся по планированию и организации самостоятельной работы.

**3.1.3** **Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

**3.1.3.1. Методика проведения текущего контроля успеваемости и промежуточной аттестации**

Контроль успеваемости и промежуточная аттестация проводятся в письменно-устной форме по списку вопросов из 3.1.4 (или тем из 2.2) , обучающемуся выдаётся не более 3 (трёх) вопросов из списка вопросов или 2 (двух тем) – далее «билет» – для подготовки письменной части ответа. Время подготовки ответа на выданные вопросы составляет не менее 1 академического часа. Использование конспектов и учебников, а также электронных устройств хранения, обработки или передачи информации при подготовке и ответе на вопросы запрещено. В случае обнаружения факта использования недозволенных материалов (устройств) составляется акт, и обучающийся удаляется с экзамена, выставляется оценка “неудовлетворительно”.

После ответа на основные вопросы билета, преподаватель вправе задать уточняющие вопросы по услышанному. Затем преподаватель задает дополнительные вопросы по любой теме из списка вопросов, вынесенных на экзамен. В качестве дополнительных используются вопросы, не требующие длительного ответа, в том числе, основные определения и понятия. Рекомендуется задавать 3-4 дополнительных вопроса на билет и не более 5 уточняющих вопросов. Преподаватель вправе увеличить количество вопросов в случае, если у него не возникает понимание, освоил обучающийся материал учебного курса или нет.

По желанию преподавателя на экзамен допустимо приглашать других преподавателей с квалификацией не ниже изложенной в п. 3.2.1 как для независимого оценивания ответов обучающихся, так и для коллегиального. В последнем случае оценка за экзамен ставится на основании голосования простого большинства. В спорных ситуациях преподаватель, ведущий дисциплину, имеет право принятия окончательного решения.

**3.1.3.2. Критерии оценивания итогового процента освоения дисциплины**

Полнота и качество ответа оцениваются по процентной шкале от 0% (нет ответа) до 100% (очень хороший ответ), результирующая оценка вычисляется следующим образом.

Полнота и качество ответов на “билет” и уточняющие вопросы усредняются, результат усреднения делится на 2. Полнота и качество ответов на дополнительные вопросы усредняются, результат усреднения делится на 2 и складывается с оценкой, полученной за ответ на “билет” и уточняющие вопросы.

Оценка за экзамен выставляется следующим образом

|  |  |  |
| --- | --- | --- |
| Полнота и качество ответов | Оценка ECTS | Оценка СПбГУ |
| 90%-100% | A | отлично |
| 80%-89% | B | хорошо |
| 70%-79% | C | хорошо |
| 61%-69% | D | удовл. |
| 50%-60% | E | удовл. |
| менее 50% | F | неуд. |

При критериях оценивания «зачёт»/«незачёт» используются критерии оценивания, описанные выше, но «неудовлетворительно» соответствует «незачёт», а остальным положительным оценкам («удовлетворительно», «хорошо», «отлично») соответствует «зачёт».

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

Пример вопросов для зачёта:

1. Программная инженерия. Предметная область. Определение. Отличие от других инженерных специальностей. Кодекс этики АСМ (ACM code of ethics).

2. Сетевой этикет. Персональные данные (понятие и виды персональных данных по ФЗ-152 Закон об охране персональных данных).

3. Информация. Понятие синтаксиса и семантики. Свойства информации. Количество информации. Юникод.

4. Операционные системы. Виды, задачи, принципиальное устройство. Понятие ядра ОС (функции ядра).

5. Виртуальная память. Полезные свойства для прикладного программиста и принципиальное устройство. Адресное пространство процесса.

6. Транслятор. Интерпретатор. Компилятор. JIT-компилятор. Оптимизации кода (языковые и платформозависимые). Основные этапы построения исполнимого модуля из исходного текста программы. Роль и принцип работы редактора связей (linker) при раздельной трансляции.

7. Системы контроля версий. Виды, история развития. Решаемые задачи. Работа с git и GitHub (git workflow, branch per feature).

8. Данные. Представление данных в памяти ЭВМ: целые числа (3 способа представления чисел со знаком). Двоично-десятичные. Арифметика с насыщением.

9. Представление данных в памяти ЭВМ: представление вещественных с плавающей точкой, специальные значения (NaN, Inf).

10. Арифметика с фиксированной точкой. Точность и ошибки в вычислениях в сравнении с float/double.

11. Представление данных в памяти ЭВМ: литералы (символы и строки). Кодировки (code pages). ASCII-Z. UTF-8.

12. (Язык С) Типы. Представление значений в памяти ЭВМ. Адресная арифметика. Приведение типов. Порядок приведения типов в выражениях.

13. (Язык С) Составные типы: объединения, структуры и массивы. Битовые поля в структурах. Размещение в памяти, выравнивание (alignment).

14. (Язык С). Операции над целочисленными значениями. Виды, семантика. Примеры.

15. (Язык С) Основные конструкции управления вычислениями. Циклы с пред- и с пост-условием, цикл for, операторы goto и switch. Графическое изображение (блок-схема, граф потока управления).

16. Способы размещения данных: глобальные и локальные переменные, область видимости, «время жизни».

17. (Язык С) Процедуры. Параметры. Локальные переменные. Организация кадра.

18. Выделение памяти: метод двоичных близнецов. Понятие внутренней и внешней фрагментации.

19. Выделение памяти блоками фиксированного размера. SLAB

20. Асимптотическая сложность. Сравнительный анализ алгоритмов и структур данных на примере различных коллекций.

21. Амортизационный анализ на примере динамически расширяемого массива.

22. Понятие хеш-функции. Область применения хеширования. Криптографическая хеш-функция. Примеры современных алгоритмов. Идеальное минимальное хеширование: определение и построение для графа.

23. Постановка задачи сортировки. Алгоритмы с квадратичной асимптотикой по времени. Устойчивые и неустойчивые сортировки.

24. Сортировки слиянием, quick sort, heap sort.

25. Атаки по радужным таблицам. Криптографическая соль.

26. Ассоциативный массив. Реализация с помощью хеш-таблицы. Разрешение коллизий: метод цепочек и прямая адресация. Абстрактный тип данных.

27. Симметричное и асимметричное шифрование: принципы организации, преимущества и недостатки. Алгоритмы шифрования DES и 3DES. Алгоритм шифрования RSA.

28. Постановка экспериментов: повторяемость, воспроизводимость. Достоверность результата.

Пример вопросов для экзамена (включает темы с зачёта для полного контроля усвоения знаний по курсу), но вопросы переформулированы и перегруппированы.

1) Программная инженерия. Предметная область. Определение. Отличие от других инженерных специальностей. Кодекс этики АСМ (ACM code of ethics).

2) Сетевой этикет. Персональные данные (понятие и виды персональных данных по ФЗ-152 «Закон об охране персональных данных»).

3) Информация. Понятие синтаксиса и семантики. Свойства информации. Количество информации. Юникод.

4) Операционные системы. Виды, задачи, принципиальное устройство. Понятие ядра ОС (функции ядра).

5) Виртуальная память. Полезные свойства для прикладного программиста и принципиальное устройство. Адресное пространство процесса.

6) Транслятор. Интерпретатор. Компилятор. JIT-компилятор. Оптимизации кода (языковые и платформозависимые). Основные этапы построения исполнимого модуля из исходного текста программы. Роль и принцип работы редактора связей (linker) при раздельной трансляции.

7) Системы контроля версий. Виды, история развития. Решаемые задачи. Работа с git и GitHub (git workflow, branch per feature).

8) Данные. Представление данных в памяти ЭВМ: целые числа (3 способа представления чисел со знаком). Двоично-десятичные. Арифметика с насыщением.

9) Представление данных в памяти ЭВМ: представление вещественных с плавающей точкой, специальные значения (NaN, Inf).

10) Арифметика с фиксированной точкой. Точность и ошибки в вычислениях в сравнении с float/double.

11) Представление данных в памяти ЭВМ: литералы (символы и строки). Кодировки (code pages). ASCII-Z. UTF-8.

12) (Язык С) Типы. Представление значений в памяти ЭВМ. Адресная арифметика. Приведение типов. Порядок приведения типов в выражениях.

13) (Язык С) Составные типы: объединения, структуры и массивы. Битовые поля в структурах. Размещение в памяти, выравнивание (alignment).

14) (Язык С). Операции над целочисленными значениями. Виды, семантика. Примеры.

15) (Язык С) Основные конструкции управления вычислениями. Циклы с пред- и с пост условием, цикл for, операторы goto и switch. Графическое изображение (блок-схема, граф потока управления).

16) Способы размещения данных: глобальные и локальные переменные, область видимости, «время жизни».

17) (Язык С) Процедуры. Параметры. Локальные переменные. Организация кадра.

18) Выделение памяти: метод двоичных близнецов. Понятие внутренней и внешней фрагментации.

19) Выделение памяти блоками фиксированного размера. SLAB

20) Асимптотическая сложность. Сравнительный анализ алгоритмов и структур данных на примере различных коллекций.

21) Амортизационный анализ на примере динамически расширяемого массива.

22) Понятие хеш-функции. Область применения хеширования. Криптографическая хеш-функция. Примеры современных алгоритмов. Идеальное минимальное хеширование: определение и построение для графа.

23) Постановка задачи сортировки. Алгоритмы с квадратичной асимптотикой по времени. Устойчивые и неустойчивые сортировки.

24) Сортировки слиянием, quick sort, heap sort.

25) Атаки по радужным таблицам. Криптографическая соль.

26) Ассоциативный массив. Реализация с помощью хеш-таблицы. Разрешение коллизий: метод цепочек и прямая адресация. Абстрактный тип данных.

27) Симметричное и асимметричное шифрование: принципы организации, преимущества и недостатки. Алгоритмы шифрования DES и 3DES. Алгоритм шифрования RSA. Цифровая подпись.

28) Постановка экспериментов: повторяемость, воспроизводимость. Достоверность результата.

29) «Управление памятью». Проблемы и решения. (Модель памяти)

30) «Мусор» и «сборка мусора». Счётчик ссылок. Модификация, работающая для кольцевого списка

31) «Трёхцветный» алгоритм «сборки мусора». Виды/варианты алгоритмов сборки мусора. Алгоритм Боэма(Boehm GC).

32) Понятие алгоритма. Машина Тьюринга. Тезис Чёрча-Тьюринга. Альтернативные формализации алгоритма (процесс вычисления, понятие данных). Теорема Райса. Исследования «оснований математики» и их влияние на развитие информатики

33) Абстрактный тип данных. Интерфейс vs реализация. Пример стека и очереди с различными способами хранения данных

34) Понятие парадигмы программирования. Основные парадигмы и хронология их появления (языки, примеры)

35) Императивная, процедурная и структурная парадигма. Теорема Бёма-Якопини.

36) Объектно-ориентированное проектирование. Основные понятия. Терминология объектно-ориентированных языков программирования.

37) Абстракции и приёмы ООП, решаемые задачи: наследование, агрегация, композиция. Примеры. Абстрактные классы. Абстрактные и виртуальные методы. Интерфейсы. Множественное наследование.

38) Полиморфизм. Полиморфизм в ООП . Вариативность. Раннее и позднее связывание. Статический полиморфизм.

39) Принципы SOLID объектно-ориентированного проектирования

40) Модель приложения MVC: компоненты и их назначение

41) Универсальная машина Тьюринга. Проблема останова.

42) Интеллектуальная собственность. Авторские права. Свободное и бесплатное ПО. Лицензии GNU GPL, BSD(MIT) и Creative Commons

43) Управление качеством в процессе создания программного продукта. Виды тестирования. Методологии TDD и BDD.

44) Код Хэмминга

45) Двоичное дерево поиска. Алгоритмы поиска, вставки, удаления, их сложность

46) Красно-чёрное дерево. Основные свойства. Теорема о максимальной высоте.

47) Повороты красно-чёрного дерева. Алгоритм добавления вершины в красно-чёрное дерево

48) АВЛ-дерево. Свойства, формулировка теоремы о высоте, идея алгоритма балансировки. АВЛ-дерево и красно-чёрное дерево: сравнительный анализ

49) Декартово дерево: возможности и границы применимости.

50) Обход графа в глубину и в ширину. Алгоритм Дейкстры поиска кратчайшего пути в графе.

51) Алгоритм Форда-Беллмана поиска кратчайшего пути в графе.

|  |  |  |
| --- | --- | --- |
| № | Код индикатора и индикатор достижения универсальной компетенции | Контрольно-измерительные материалы (КИМ) (тестовые вопросы, контрольные задания, кейсы и пр.) |
|  | 1 | 2 |
| 1 | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения | Ответ на каждый вопрос билетов и на дополнительные вопросы, на которые обучающийся отвечает на зачёте и экзамене, оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 2 | ОПК-2.1 Уметь писать программный код с использованием языков программирования, определения и манипулирования данными | Ответ на каждый вопрос билетов и на дополнительные вопросы, на которые обучающийся отвечает на зачёте и экзамене, оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 3 | ОПК-6.1 Уметь проверять и отлаживать программный код | Во время ответов на вопрос 20 или 21 зачёта или во время ответов на вопросы 20, 21, 45-51 экзамена или во время ответов на дополнительные вопросы по этим темам знания и умения обучающегося по данной компетенции оцениваются по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100. |
| 4 | ПКА-1.1 Быть способным осуществлять организационное и технологическое обеспечение кодирования на языках программирования | Ответ на каждый вопрос билетов и на дополнительные вопросы, на которые обучающийся отвечает на зачёте и экзамене, оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 5 | ПКП-1.1 Уметь разрабатывать технических спецификаций на программные компоненты и их взаимодействие | Ответ на каждый вопрос билетов и на дополнительные вопросы, на которые обучающийся отвечает на зачёте и экзамене, оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 6 | ПКП-2.1 Уметь описывать алгоритмы компонентов, включая методы и схемы | Ответ на каждый вопрос билетов и на дополнительные вопросы, на которые обучающийся отвечает на зачёте и экзамене, оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 7 | УК 1.4. Оценивает достоинства, недостатки и последствия вариантов решения поставленных задач; | Ответ на каждый вопрос билетов и на дополнительные вопросы, на которые обучающийся отвечает на зачёте и экзамене, оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 8 | УК-3.3. Строит продуктивное взаимодействие с учетом возможных последствий личных действий в социальном взаимодействии и командной работе; | Ответ на каждый вопрос билетов и на дополнительные вопросы, на которые обучающийся отвечает на зачёте и экзамене, оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем образовании по соответствующему направлению.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Специальных требований нет.

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

Учебные аудитории для проведения учебных занятий, оснащенные стандартным оборудованием, используемым для обучения в СПбГУ в соответствии с требованиями материально-технического обеспечения

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Стандартное оборудование, используемое для обучения в СПбГУ. MS Windows, MS Office, Mozilla FireFox, Google Chrome, Acrobat Reader DC, WinZip, Антивирус Касперского

**3.3.3 Характеристики специализированного оборудования**

Специальных требований нет.

**3.3.4 Характеристики специализированного программного обеспечения**

Специальных требований нет.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Специальных требований нет.

**3.4. Информационное обеспечение**

**3.4.1 Список литературы**

1. Керниган, Б. В. Язык программирования C : учебник / Б. В. Керниган, Д. М. Ричи. — 2-е изд. — Москва : ИНТУИТ, 2016. — 313 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100543

2. Брукс Ф. Мифический человеко-месяц или как создаются программные системы.- СПб, 2007. – 298 с. Мм – 10 экз.

3. Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. — Санкт-Петербург : Лань, 2019. — 700 с. — ISBN 978-5-8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система (доступно по подписке СПбГУ). — URL: https://e.lanbook.com/book/119661

4. Сомон., П. И. Волшебство Kotlin : руководство / П. И. Сомон. ; перевод с английского А. Н. Киселева.. — Москва : ДМК Пресс, 2020. — 536 с. — ISBN 978-5-97060-801-2. — Текст : электронный // Лань : электронно-библиотечная система (доступно по подписке СПбГУ). — URL: https://e.lanbook.com/book/140599

5. Жемеров, Д. Kotlin в действии / Д. Жемеров, С. Исакова ; перевод с английского А. Н. Киселев. — Москва : ДМК Пресс, 2018. — 402 с. — ISBN 978-5-97060-497-7. — Текст : электронный // Лань : электронно-библиотечная система (доступно по подписке СПбГУ). — URL: <https://e.lanbook.com/book/112926>

6. Довек, Ж. Введение в теорию языков программирования / Ж. Довек, Ж. -. Леви. — Москва : ДМК Пресс, 2013. — 134 с. — ISBN 978-5-94074-913-4. — Текст : электронный // Лань : электронно-библиотечная система (доступно по подписке СПбГУ). — URL: <https://e.lanbook.com/book/82826>

**3.4.3 Перечень иных информационных источников**

• Сайт Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/>

• Электронный каталог Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/cgi-bin/irbis64r/cgiirbis_64.exe?C21COM=F&I21DBN=IBIS&P21DBN=IBIS>

• Перечень электронных ресурсов, находящихся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/>

• Перечень ЭБС, на платформах которых представлены российские учебники, находящиеся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/browse?name=rures&resource%20type=8>

**Раздел 4. Разработчики программы**

Кириленко Яков Александрович нет нет cт. преподаватель каф. системного программирования [y.kirilenko@spbu.ru](https://mail.spbu.ru/Session/3175900-QxOfHCC4kGpgJBlfFjiZ/Message.wssp?Mailbox=Кафедра%20системного%20программирования&MSG=11)