**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Программирование на языке C

Programming C

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 051473

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Обучение обучающихся программированию на языке С. Уметь использовать язык С для профессиональной деятельности, в т.ч. для выполнения алгоритмических расчетов.

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

Общее среднее образование.

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

Уметь использовать методы и инструментальные средства исследования объектов профессиональной деятельности.

Знать содержание дисциплины «Программирование на С» и иметь достаточно полное представление о возможностях применения её разделов для свое профессиональной деятельности.

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

Практические занятия 20 часов

**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 2 |  |  | 2 | 30 |  |  |  |  | 2 |  | 15 |  | 31 |  | 28 |  | 20 | 3 |
|  |  |  | 1-25 | 1-12 |  |  |  |  | 1-25 |  | 1-1 |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО |  |  | 2 | 30 |  |  |  |  | 2 |  | 15 |  | 31 |  | 28 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 обученияочная | | | | | | |
| Семестр 2 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

1. Программа «Hello World!».
2. Утилита «make». Инструкции и операторы. Приоритет операторов. Массивы. Строки. Указатели и ссылки. Функции. Комментарии. Циклы.
3. Структуры.
4. Макросы. Директива «Include». Обзор стандартных библиотек.
5. Работа с файлами.

Операции ввода/вывода. Работа с файлами.

1. «Python.h»
2. Написание библиотек на С для скриптов, написанных на Python.
3. Сортировки.

Реализация алгоритмов основных сортировок: «Bubble», «Heap», «Merge», «Quick», «Distribution».

1. Структуры данных на массивах.

Реализация основных структур данных на массивах. Алгоритм обратной польской записи.

1. Поиск.

Задача поиска точек в прямоугольнике. Задача одновременного поиска в наборе упорядоченных списков. Fractional cascading.

1. Поиск порядковых статистик.

Реализация алгоритмов. Точный алгоритм Munro-Paterson, Канна-Гринвальда. Приближенный алгоритм Manku-Rajagopalan-Lindsay. Приближенный подсчет числа различных элементов. Поиск частотных элементов.

1. Деревья.

Реализация основных структур: деревья поиска, черно-красные деревья, splay-деревья, R-дерево, декартово дерево.

1. Деревья.

RMQ и Фарах-Колтона-Бендера алгоритм. LCA и алгоритм Таржана.

1. Обход дерева.

Эйлеров обход дерева (в т.ч. для структур с внешней памятью).

1. Графы.

Точки сочленения и их нахождение за линейное время. Поиск сильно связных компонент

1. Циклы графа.

Топологическая сортировка. Сильно связные компоненты. Циклы графа (поиск). Разрезы графа. Раскраска двумя цветами. Алгоритмы Форда-Беллмана и Флойда. Алгоритм Дейкстры (в т.ч. двунапревленный). Алгоритм ALT.

1. Остовное дерево.

Алгоритмы Крускала, Примы и Борувки (взвешенные графы). Алгоритм Борувки (во внешней памяти).

1. Гамильтонов цикл

Поиск гамильтонова цикла.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Посещение занятий, выполнение самостоятельных работ.

**3.1.2 Методическое обеспечение самостоятельной работы**

Не предусмотрено.

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

В течение семестра по дисциплине проводятся аудиторные контрольные работы, проводится экзамен.

Контрольная работа состоит из 1–5 задач, за каждую из которых обучающийся может получить от минимального (написана явная чушь, демонстрирующая полное непонимание происходящего обучающимся, в том числе орфографические ошибки в написании важнейших ключевых слов языка программирования C, и т. п.) до максимального (использованный алгоритм верен и эффективен, отсутствуют синтаксические и семантические ошибки при записи этого алгоритма на языке программирования C, обучающийся демонстрирует способность к проектированию алгоритмов и свободное владение этим языком программирования в объёме, необходимом для решения данной задачи). Максимальное количество баллов указывается в условии каждой задачи. Максимальный вклад в итоговый уровень усвоения дисциплины каждой контрольной работы одинаков, и максимальный общий вклад всех контрольных работ составляет 50% и вклад экзамена — оставшиеся 50%.

Экзамен проводится в устной форме. Обучающийся получает три вопроса, максимальное время подготовки — не более часа, ответ на каждый из них оценивается от -максимального (написана явная чушь, демонстрирующая полное непонимание происходящего обучающимся, в том числе орфографические ошибки в написании важнейших ключевых слов языка программирования C, и т. п.) до максимального (ответ является верным и исчерпывающим, обучающийся демонстрирует способность к проектированию алгоритмов и свободное владение языком программирования C в объёме, необходимом для ответа на данный вопрос). Вклад этих оценок в итоговый уровень освоения дисциплины 15%. Оставшиеся 5% приходятся на оценку ответов на дополнительные вопросы от -максимального (обучающийся демонстрирует полное непонимание происходящего) до максимального (обучающийся отвечает на все дополнительные вопросы быстро, кратко и правильно).

Если итоговый уровень усвоения дисциплины получается отрицательным, для расчета окончательной оценки он принимается нулевым. Окончательная оценка выставляется, исходя из уровня по следующему правилу: от 91% до 100% -- оценка A (отлично), от 81% до 90% -- оценка B (хорошо), 71% до 80% -- оценка C (хорошо), 61% до 70% -- оценка D (удовлетворительно), 51% до 60% -- оценка E (удовлетворительно), 50% и менее -- оценка F (неудовлетворительно).

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

Примерный перечень вопросов к экзамену на текущий период обучения:

1. Программа «Hello world!»
2. Структуры.
3. Работа с файлами.
4. «Python.h».
5. Сортировки.
6. Структуры данных на массиве.
7. Поиск.
8. Поиск порядковых статистик.
9. Деревья.
10. Обход дерева.
11. Графы.
12. Циклы графа.
13. Остовное дерево.
14. Гамильтонов цикл.

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

Оценка обучающимися содержания и качества учебного процесса по дисциплине осуществляется в установленном в СПбГУ порядке.

**3.2. Кадровое обеспечение**

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

К чтению лекций должны привлекаться преподаватели, имеющие ученую степень кандидата или доктора наук (в том числе степень PhD, прошедшую установленную процедуру признания и установления эквивалентности). Преподаватели, привлекаемые к проведению практических занятий, должны иметь базовое образование и/или ученую степень, соответствующие профилю преподаваемой дисциплины.

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

Для подготовки и проведения занятий требуется лаборант или инженер для следующих работ: 1) техническая подготовка каталогов исходных данных в форме, удобной для учебной работы; 2) поддержания работоспособности компьютерного класса.

**3.3. Материально-техническое обеспечение**

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

Компьютерный класс.

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

Компьютерный класс. Компьютеры под управлением операционной системы семейства Linux, с установленным интерпретатором Python.

**3.3.3 Характеристики специализированного оборудования**

Не предусматриваются.

**3.3.4 Характеристики специализированного программного обеспечения**

Не предусматриваются.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Не предусматриваются.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. Ivor Horton. Beginning C. From Novice to Professional. - APress, 2006. ЭР по подписке СПбГУ: https://find.library.spbu.ru/vufind/Record/978-1-4302-0243-1.

2. Jeri N. Hanly, Elliot B. Koffman. Problem Solving and Program Design in C. - Pearson Education, 2007. ЭР открытого доступа в сети Интернет.

3. Kyle Loudon. Mastering Algorithms with C. - O'Really, 1999. - ЭР открытого доступа в сети Интернет.

**3.4.2 Список дополнительной литературы**

1. Керниган Б., Д. Ритчи. Язык программирования Си. – СПб: Невский Диалект, 2001-2013.

**3.4.3 Перечень иных информационных источников**

Не предусмотрено.

**Раздел 4. Разработчики программы**

Ананьевский Михаил Сергеевич, кандидат физ.-мат. наук, доцент, m.s.ananyevskiy@spbu.ru.