**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: 7

Регистрационный номер рабочей программы: 051476

2020

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Обучение обучающихся основным методам программирования на языке C++, подготовка студентов к восприятию других дисциплин, использующих систему понятий основных парадигм программирования, а также к использованию этих методов при решении прикладных и теоретических задач; развитие у студентов доказательного, логического мышления, подготовка к самостоятельным научным исследованиям; подготовка к восприятию других дисциплин программирования.

Поставленные цели достигаются путём решения следующих задач курса: изучение основных парадигм программирования, предоставляемых языком C++; развитие навыков самостоятельного решения практических задач и интерпретации полученных результатов; обеспечение базы для усвоения современных методов программирования и использования соответствующих инструментальных наборов; повышение математической вообще и программистской в частности культуры обучающегося.

Курс «Программирование на C++» дает обучающимся комплекс современных методов, позволяющих проектировать и реализовывать программы широкого спектра приложений. Дисциплина «Программирование на C++» является одной из базовых в подготовке к профессиональной деятельности в области информационных технологий и служит основой для изучения других программистских дисциплин как теоретического, так и прикладного характера, входящих в программу обучения на факультете.  
  
**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Для успешного освоения дисциплины обучающийся должен иметь предварительную подготовку по дисциплинам «Введение в современное программирование» и «Программирование на C», изучаемым на I курсе математико-механического факультета университета.

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

Выпускник должен знать содержание дисциплины «Программирование на C++» и иметь представление о возможностях применения ее разделов, уметь применять на практике различные парадигмы программирования, поддерживаемые языком C++, уметь выбирать правильную парадигму (или их комбинацию) для решения конкретной задачи.  
  
  
**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Аудиторная учебная работа: практические занятия. Контрольные работы. Самостоятельная работа с использованием методических материалов: индивидуальная работа с рекомендованной основной и дополнительной литературой по программированию на языке С++.

**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 3 |  |  |  | 30 |  |  |  |  | 2 | |  |  | 28 | | 4 | |  | 8 |  | 30 | 2 |
|  |  |  |  | 1-12 |  |  |  |  | 1-12 | |  |  | 1-12 | | 1-1 | |  | 1-1 |  |  |  |
| Семестр 4 |  |  |  | 30 |  |  |  |  | 2 | |  |  | 30 | | 4 | |  | 6 |  | 30 | 2 |
|  |  |  |  | 1-12 |  |  |  |  | 1-12 | |  |  | 1-12 | | 1-1 | |  | 1-1 |  |  |  |
| Семестр 5 |  |  | 2 | 30 |  |  |  |  | 2 | |  |  | 30 | | 36 | |  | 8 |  | 30 | 3 |
|  |  |  | 1-100 | 1-12 |  |  |  |  | 1-25 | |  |  | 1-12 | | 1-1 | |  | 1-1 |  |  |  |
| ИТОГО |  |  | 2 | 90 |  |  |  |  | 6 | |  |  | 88 | | 44 | |  | 22 |  |  | 7 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 3 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |
| Семестр 4 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |
| Семестр 5 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

**3 семестр**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| 1 | Процедурное программирование на C++ | Практические занятия | 6 |
| Сам. раб. в присутствии преп. | 6 |
| По методическим материалам |  |
| 2 | Объектно-ориентированное программирование на C++ | Практические занятия | 10 |
| Сам. раб. в присутствии преп. | 10 |
| По методическим материалам |  |
| 3 | Обобщенное программирование на C++ | Практические занятия | 14 |
| Сам. раб. в присутствии преп. | 12 |
| По методическим материалам | 4 |
|  | Промежуточная аттестация | Сам. раб. | 8 |
| Зачет | 2 |
| **Итого** | | | **72** |

**4 семестр**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| 4 | Функциональное программирование на C++ | Практические занятия | 6 |
| Сам. раб. в присутствии преп. | 6 |
| По методическим материалам |  |
| 5 | Параллельное программирование на C++ | Практические занятия | 14 |
| Сам. раб. в присутствии преп. | 14 |
| По методическим материалам | 4 |
| 6 | Регулярные выражения, функции времени и даты, генераторы случайных чисел как часть стандартной библиотеки C++ | Практические занятия | 10 |
| Сам. раб. в присутствии преп. | 10 |
| По методическим материалам |  |
|  | Промежуточная аттестация | Сам. раб. | 6 |
| Зачет | 2 |
| **Итого** | | | **72** |

**5 семестр**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| 7 | Вопросы безопасности исключений | Практические занятия | 8 |
| Сам. раб. в присутствии преп. | 8 |
| По методическим материалам | 12 |
| 8 | Вопросы косвенных обращений и мимикрии | Практические занятия | 10 |
| Сам. раб. в присутствии преп. | 10 |
| По методическим материалам | 12 |
| 9 | Образцы проектирования | Практические занятия | 12 |
| Сам. раб. в присутствии преп. | 12 |
| По методическим материалам | 12 |
|  | Промежуточная аттестация | Сам. раб. | 8 |
| Консультации | 2 |
| Экзамен | 2 |
| **Итого** | | | **108** |

**Содержание учебных занятий:**

**Тема 1.** *Процедурное программирование на C++* (6 часов практических занятий, 6 часов самостоятельной работы в присутствии преподавателя).

Наследство языка C: общая структура программы, препроцессор, выражения, типы, операции, математические функции, операторы, пользовательские функции, массивы, указатели, структурные типы данных. Новшества C++ в плане процедурного программирования: цикл для перебора элементов массива, пространства имен, исключения, потоковый ввод/вывод, сырые литералы.

**Тема 2.** *Объектно-ориентированное программирование на C++* (10 часов практических занятий, 10 часов самостоятельной работы в присутствии преподавателя).

Поддержка объектно-ориентированного программирования в C++: методы структурных типов данных, в том числе конструкторы и деструкторы, инициализаторы, константные объекты и методы, статические поля и методы, права доступа, классы, друзья, наследование, права доступа при наследовании, множественное наследование, виртуальные базовые классы и тонкости инициализации, полиморфизм (через наследование) и диспетчеризация, в том числе двойная.

**Тема 3.** *Обобщенное программирование на C++* (14 часов практических занятий, 12 часов самостоятельной работы в присутствии преподавателя, 4 часа по методическим материалам).

Перегрузка функций, методов и операций. Наследство языка C: функции с неизвестным заранее числом и типами параметров. Шаблонные функции и классы, шаблонные друзья и методы с дополнительными шаблонными параметрами. Возможные виды шаблонных параметров. Специализации шаблонов. Вложенные классы. Понятие итератора, виды итераторов. Классы контейнеров и итераторов стандартной библиотеки. Основные алгоритмы, цикл for для перебора содержимого контейнера без явного использования итераторов. Ключевое слово auto для типов переменных и результатов функций. Явное инстанциирование шаблонов. Начала метапрограммирования с использованием шаблонов, ключевое слово constexpr как средство упрощения несложных метапрограмм. Реализация отложенных вычислений. Литералы пользовательских типов.

**Тема 4.** *Функциональное программирование на C++* (6 часов практических занятий, 6 часов самостоятельной работы в присутствии преподавателя).

Понятие указателя на функцию, его использование для передачи функции в качестве параметра в другую функцию. Указатели на поля и методы класса. Безымянные функции: их синтаксис и типы. Подводные камни использования безымянных функций в C++. Средства для получения новых функций путем подстановки констант вместо некоторых из параметров уже имеющихся функций, в том числе для методов. Использование массивов и контейнеров стандартной библиотеки, в том числе ассоциативных, для хранения таких указателей. Построение шаблонных функций с использованием шаблонов, допускающих различное количество параметров.

**Тема 5*.***  *Параллельное программирование на C++*. (14 часов практических занятий, 14 часов самостоятельной работы в присутствии преподавателя, 4 часа по методическим материалам)

Параллелизм из стандарта C++ как существенно более мощный и удобный вариант параллелизма, чем OpenMP. «Дешевый» параллелизм через механизм будущих результатов и примитив асинхронного выполнения функций. Явное управление потоками через методы класса, представляющего потоки в программе. Блокирующая и безблокировочная синхронизация, мьютексы и замки. Модель памяти языка C++, атомарные типы, в том числе шаблонные. Проектирование параллельных структур данных.

**Тема 6.** *Регулярные выражения, функции времени и даты, генераторы случайных чисел как часть стандартной библиотеки C++* (10 часов практических занятий, 10 часов самостоятельной работы в присутствии преподавателя).

Понятие о регулярных выражениях. Использование регулярных выражений для распознавания определенных видов строк, а также для извлечения из них определенной информации, или для их модификации. Использование функций для обработки даты и времени, а также некоторых возможностей генераторов случайных чисел из состава стандартной библиотеки C++.

**Тема 7.** *Вопросы безопасности исключений* (8 часов практических занятий, 8 часов самостоятельной работы в присутствии преподавателя).

Понятие программы, безопасной по отношению к исключениям. Различные уровни гарантий такой безопасности. Основные средства языка C++ и его стандартной библиотеки для написания программ, безопасных по отношению к исключениям. Общие рекомендации написания таких программ. Конкретные примеры программ, небезопасных по отношению к исключениям, и результаты их исправления.

**Тема 8*.***  *Вопросы косвенных обращений и мимикрии*. (10 часов практических занятий, 10 часов самостоятельной работы в присутствии преподавателя)

Понятие о косвенных обращениях и мимикрии в C++. Классы умных указателей с различной функциональностью (проверка разыменования nullptr, значения полей вычисляются в момент разыменования, подгрузка информации с диска при разыменовании, семантика подсчета ссылок и т. д.) Классы-обертки для указателей, использующих полиморфизм, указатели-хозяева своих объектов (средства из стандартной библиотеки и методы их реализации). Классы, объекты которых притворяются функциями (одно- и двухместные предикаты, другие операции), и стандартные алгоритмы, использующие такие объекты.

**Тема 9.** *Образцы проектирования*(12 часов практических занятий, 12 часов самостоятельной работы в присутствии преподавателя).

Понятие об образцах проектирования как о стандартных решениях типовых задач проектирования. Примеры конкретных образцов проектирования: Абстрактная фабрика, Строитель, Отложенная инициализация, Пул одиночек, Фабричный метод, Объектный пул, Прототип, Получение ресурса есть инициализация, Одиночка, Адаптер, Мост и т. д.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Обучающиеся должны посещать практические занятия, выполнять задания преподавателей.

Обучающемуся необходимо знать содержание практических занятий, уметь применять полученные знания при решении конкретных практических задач.

**3.1.2 Методическое обеспечение самостоятельной работы**

При самостоятельном выполнении домашних, индивидуальных и контрольных заданий целесообразно использовать рекомендованные учебники и задачники, а также дополнительную литературу.

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

В течение каждого семестра по дисциплине проводятся аудиторные контрольные работы, проводятся зачеты (3-й и 4-й семестр) и экзамен (5-й семестр).

Контрольная работа состоит из 1–5 задач, за каждую из которых обучающийся может получить 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 (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 60% для тех семестров, где проводится зачет (его вклад — оставшиеся 40%) и 50% для тех, где проводится экзамен (его вклад — оставшиеся 50%).

Зачет проводится в устной форме. Обучающийся получает два вопроса, максимальное время подготовки — не более получаса, 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 10% 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).

Экзамен проводится в устной форме. Обучающийся получает три вопроса, максимальное время подготовки — не более часа, ответ на каждый из них оценивается 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 (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 список вопросов для проведения зачета (3 семестр):*

1. Общая структура программы на C++.
2. Препроцессор.
3. Выражения, типы, операции, математические функции.
4. Операторы.
5. Пользовательские функции.
6. Массивы.
7. Указатели.
8. Структурные типы данных.
9. Пространства имен.
10. Исключения.
11. Потоковый ввод/вывод.
12. Методы структурных типов данных.
13. Конструкторы и деструкторы, инициализаторы.
14. Константные объекты и методы.
15. Статические поля и методы.
16. Права доступа, классы, друзья.
17. Наследование, права доступа при наследовании, множественное наследование.
18. Виртуальные базовые классы и тонкости инициализации.
19. Полиморфизм (через наследование) и диспетчеризация.
20. Двойная диспетчеризация.
21. Перегрузка функций, методов и операций...
22. Функции с неизвестным заранее числом и типами параметров (как в C).
23. Шаблонные функции и классы, шаблонные друзья и методы с дополнительными шаблонными параметрами.
24. Возможные виды шаблонных параметров. Специализации шаблонов. Вложенные классы.
25. Понятие итератора, виды итераторов.
26. Классы контейнеров и итераторов стандартной библиотеки.
27. Основные алгоритмы, цикл for для перебора содержимого контейнера без явного использования итераторов.
28. Ключевое слово auto для типов переменных и результатов функций. Явное инстанциирование шаблонов.
29. Начала метапрограммирования с использованием шаблонов, ключевое слово constexpr как средство упрощения несложных метапрограмм.
30. Реализация отложенных вычислений. Литералы пользовательских типов.

*Примерный список вопросов для проведения зачета (4 семестр):*

1. Понятие указателя на функцию, его использование для передачи функции в качестве параметра в другую функцию.
2. Указатели на поля и методы класса.
3. Безымянные функции: их синтаксис и типы. Подводные камни использования безымянных функций в C++.
4. Средства для получения новых функций путем подстановки констант вместо некоторых из параметров уже имеющихся функций, в том числе для методов.
5. Использование массивов и контейнеров стандартной библиотеки, в том числе ассоциативных, для хранения указателей на поля и методы класса.
6. Построение шаблонных функций с использованием шаблонов, допускающих различное количество параметров.
7. Параллелизм из стандарта C++ как существенно более мощный и удобный вариант параллелизма, чем OpenMP.
8. «Дешевый» параллелизм через механизм будущих результатов и примитив асинхронного выполнения функций.
9. Явное управление потоками через методы класса, представляющего потоки в программе.
10. Блокирующая и безблокировочная синхронизация, мьютексы и замки.
11. Модель памяти языка C++, атомарные типы, в том числе шаблонные.
12. Проектирование параллельных структур данных.
13. Понятие о регулярных выражениях.
14. Использование регулярных выражений для распознавания определенных видов строк.
15. Использование регулярных выражений для извлечения из строк определенной информации, или для их модификации.
16. Использование функций для обработки даты и времени.
17. Возможности генераторов случайных чисел из состава стандартной библиотеки C++ и их использование.

*Примерный перечень вопросов к экзамену (5 семестр):*

1. Понятие программы, безопасной по отношению к исключениям. Различные уровни гарантий такой безопасности.
2. Основные средства языка C++ и его стандартной библиотеки для написания программ, безопасных по отношению к исключениям.
3. Общие рекомендации написания таких программ.
4. Конкретные примеры программ, небезопасных по отношению к исключениям, и результаты их исправления.
5. Понятие о косвенных обращениях и мимикрии в C++.
6. Классы умных указателей с различной функциональностью (проверка разыменования nullptr, значения полей вычисляются в момент разыменования, подгрузка информации с диска при разыменовании, семантика подсчета ссылок и т. д.)
7. Классы-обертки для указателей, использующих полиморфизм.
8. Указатели-хозяева своих объектов (средства из стандартной библиотеки и методы их реализации).
9. Классы, объекты которых притворяются функциями (одно- и двухместные предикаты, другие операции), и стандартные алгоритмы, использующие такие объекты
10. Понятие об образцах проектирования как о стандартных решениях типовых задач проектирования.
11. Примеры конкретных образцов проектирования (набор вопросов по каждому из таких образцов, список см. в википедии).

**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 преподаванию практических занятий могут быть допущены преподаватели, имеющие диплом о высшем образовании по соответствующему направлению или ученую степень к.ф.-м.н. или д.ф.-м.н. или звание доцента или профессора.

**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**

Требуются компьютерные классы.

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

Нужны персональные компьютеры (число не меньше числа студентов), не ниже процессор Core i3/тактовая частота 2ГГц/оперативная память 2Гб/жесткий диск 300Гб/оптический дисковод DVD-RW/монитор 1200х1000/сеть/клавиатура/мышь, Windows 7/8/8.1/10, Visual Studio 2013-17

**3.3.3 Характеристики специализированного оборудования**

Не предусмотрены.

**3.3.4 Характеристики специализированного программного обеспечения**

Не предусмотрены.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Не предусмотрены.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. Gregoire Marc R., Solter Nicholas, Kleper Scott. Professional C++. - Indianapolis, IN: Wiley Pubishing, Inc., 2011. ЭР по подписке СПбГУ: https://find.library.spbu.ru/vufind/Record/EBC818960

**3.4.2 Список дополнительной литературы**

1. <https://docs.microsoft.com/en-us/cpp/opbuildpdf/cpp/TOC.pdf?branch=live>

2. https://docs.microsoft.com/en-us/cpp/opbuildpdf/standard-library/TOC.pdf?branch=live

**3.4.3 Перечень иных информационных источников**

Материалы, размещенные в сети Интернет:  
 http://www.cplusplus.com, <http://www.cppreference.com/>.

**Раздел 4. Разработчики программы**

Лебединский Дмитрий Михайлович, к.ф.-м.н., доцент кафедры параллельных алгоритмов, d.lebedinsky@spbu.ru.