**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Язык программирования С++

C++ Programming Language

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 4

Регистрационный номер рабочей программы: 064364

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Дисциплина ориентирована на профессиональное и личностное становление будущих бакалавров в области информационных технологий. Основной целью освоения дисциплины «С++» является изучение парадигм современного программирования на примере языка C++, а также получение навыка разработки программ на этом языке.

Задачи курса:

1. научить студентов проектированию и разработке программ на языке C++ с использованием парадигм современного программирования (процедурного, объектно-ориентированного и обобщенного подходов);
2. научить студентов использованию стандартных библиотек (libc, STL);
3. научить студентов использованию технологий и инструментов, позволяющих производить разработку, отладку и тестирование программ на C++ на современном технологическом уровне;
4. ознакомить студентов с техническими деталями работы аппаратного обеспечения и операционных систем, понимание которых необходимо для разработки на С++.

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

Студент, обучающийся по данной дисциплине, должен иметь знания, умения и навыки, соответствующие полному среднему образованию, т.е. иметь базовые навыки алгоритмизации и разработки программ на одном из существующих языков программирования (например, Basic, Pascal или C).

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

Обучающийся должен овладеть теоретическим материалом в объеме, предусмотренном программой, уметь применять полученные знания при решении теоретических и прикладных задач, на основе анализа освоенных разделов:

1. Программа, состоящая из нескольких файлов;
2. Указатели ;
3. Три вида памяти в C++;
4. Стандартная библиотека libc;
5. Объектно-ориентированное программирование ;
6. Обобщенное программирование (шаблоны);
7. Обработка ошибок с использованием исключений;
8. Стандартная библиотека STL (контейнеры, алгоритмы);
9. Стандарты C++11 – С++17.

По результатам обучения обучающийся должен:

**Знать**

- основные языковые конструкции С++;

- механизмы процедурного программирования на C++;

- механизмы объектно-ориентированного программирования на C++;

- механизмы обобщенное программирование на C++;

- принципы обработки ошибок на C++;

- возможности библиотек libc и STL;

-средства разработки (препроцессор, компилятор, ассемблер, компоновщик, система сборки, отладчик и т.д.);

**Уметь:**

**-** разрабатывать программы для выбранной задачи на языке С++;

- писать код в соответствии с заданным стилем программирования;

- читать чужой код;

- организовывать автоматическое тестирование программ, используя юнит-тесты;

- проводить отладку программы.

**Владеть:**

- навыками разработки программы с использованием парадигм современного программирования (процедурный, объектно-ориентированный и обобщенный подходы);

- навыками создания программы с использованием современных средств разработки;

- навыками определения целесообразности применения изученных технологий (языковых конструкций С++, стандартных и сторонних библиотек) и средств разработки для решения выбранной задачи.

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

Практические занятия 30 часов, консультации 2 часа, промежуточная аттестация (экзамен) 2 часа.

**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 4 | 30 |  | 2 | 30 |  |  |  |  | 2 |  |  |  | 56 |  | 24 |  | 34 | 4 |
|  | 2-70 |  | 2-70 | 10-25 |  |  |  |  | 10-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 30 |  | 2 | 30 |  |  |  |  | 2 |  |  |  | 56 |  | 24 |  | 34 | 4 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 4 |  |  | | экзамен, устно, традиционная форма | по графику промежуточной аттестации, |  |  |

**2.2. Структура и содержание учебных занятий**

**В данной таблице указываются разделы (модули), их содержание (темы) и часы.**

*Семестр 4*

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| 1 | 1.   Шаблоны   * решение в стиле C * шаблонные классы * шаблонные функции * специализация шаблонов | лекции | 4 |
| Практические занятия | 4 |
| по методическим материалам | 6 |
| 2 | 2. Исключения   * обработка ошибок в стиле C * try/catch/throw * исключения в конструкторах и деструкторах * идиома RAII * гарантии исключений | лекции | 4 |
| практические занятия | 4 |
| по методическим материалам | 6 |
| 3 | 3.   Последовательные контейнеры   * string, vector, list, deque * array, forward\_list * внутреннее устройство и основные операции * итераторы и их инвалидация | лекции | 4 |
| Практические занятия | 4 |
| по методическим материалам | 6 |
| 4 | 4.   Ассоциативные контейнеры   * set, multiset, map, multimap * unordered\_set, unordered\_map * внутреннее устройство и основные операции * итераторы и их инвалидация | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 4 |
| 5 | 5.   Алгоритмы   * функторы * обзор алгоритмов с примерами * std::advance * iterator\_traits (value\_type, iterator\_category) | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 4 |
| 6 | 6.   Приведение типов   * static\_cast, const\_cast, reinterpret\_cast * RTTI и dynamic\_cast * mutable | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 4 |
| 7 | 7.   Move семантика   * rvalue и lvalue * rvalue references * move constructor, move assignment * std::move | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 4 |
| 8 | 8.   Переменное число аргументов   * переменное число параметров в стиле C (va\_arg, va\_list, va\_start) * variadic templates | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 4 |
| 9 | 9.   С++11. Разное.   * lambda * auto, decltype * default, delete, override, final * initializer\_list | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 4 |
| 10 | 10.  Метапрограммирование   * решение в стиле C * type\_traits * enable\_if | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 4 |
| 11 | 11.  Потоки и мутексы   * зачем нужны потоки? * конкурентный доступ к переменным * thread * mutex, lock\_guard | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 4 |
| 12 | 12.  Сondition variable   * задача "производитель-потребитель" * thread и std::bind * condition\_variable | лекции | 2 |
| Практические занятия | 2 |
| по методическим материалам | 6 |

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

В методическое обеспечение аудиторной работы включены материалы лекций, учебные пособия, в том числе электронные

*Методические рекомендации преподавателю*

Изложение материалов на занятиях следует выстраивать от простого к сложному. Новые термины должны появляться по мере усвоения предыдущих. По возможности, следует побуждать учащихся к активной деятельности на лекциях с помощью проверочных вопросов во время лекций, заданий с последующим обсуждением ответов и т.п. Для проведения лекционных занятий рекомендуется использовать презентации.

На практических занятиях активное взаимодействие со студентами должно быть основной формой проведения занятий. На каждом занятии можно сначала обсудить правильные и неправильные способы решения предыдущего домашнего задания, а затем рассматривать новую тему и предложить студентам решить примеры заданий с помощью преподавателя.

Для проведения зачета и экзамена преподаватель должен знать критерии оценивания.

*Методические указания студентам*

Самостоятельная работа студентов включает в себя решение задач, изучение лекционного материала, учебников, учебных пособий и иных материалов. Время и место самостоятельной работы выбираются студентами по своему усмотрению с учетом рекомендаций преподавателя.

После окончания лекции и до наступления следующей лекции по предмету студенту следует уделить время тому, чтобы самостоятельно пройти по всему материалу лекции и убедиться, что весь материал понятен. При необходимости, можно обратиться за разъяснениями к другим обучающимся на курсе, или к преподавателю курса (например, через средства информационной поддержки курса). После повторения материала лекции, полезно также обратиться к источникам из списка дополнительной литературы по курсу, чтобы получить более полное представление о пройденной теме.

**3.1.2 Методическое обеспечение самостоятельной работы**

В состав методического обеспечения самостоятельной работы входит основная и дополнительная литература, контрольные вопросы и задания для организации текущей самостоятельной работы обучающихся по всем учебным разделам.

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

Показателями, характеризующими текущую учебную работу слушателей, являются:

* активность посещения занятий и работы на занятиях;
* оценка домашних заданий;
* оценка тестов.

Критерии оценивания формируются исходя из данных показателей работы слушателей, и методов измерения знаний, которые используются при контроле освоения учебного материала дисциплины слушателями.

В процессе обучения студент получает баллы за тесты, недельные и семестровые домашние задания. Тесты проводятся 1 раз в месяц; недельное задание – раз в 1 или 2 недели; семестровое задание – 1 или 2 раза в семестр.

Для получения допуска к экзамену необходимо в семестре набрать 30% от максимума баллов. Экзамен проводится в форме устного ответа на вопрос и выполнения письменной задачи (написание программы). На подготовку устного ответа отводится 30 минут. На решение задачи – 1 час.

При оценивании результатов контрольных заданий, письменных работ и промежуточной (заключительной) аттестации по дисциплине преподавателю рекомендуется использовать следующие условия выставления оценок:

Оценки «отлично» заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание учебного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную литературу и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка «отлично» выставляется усвоившим взаимосвязь основных понятий дисциплины и их значение для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.

Оценки «хорошо» заслуживает студент, обнаруживший полное знание учебного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка «хорошо» выставляется учащимся, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.

Оценки «удовлетворительно» заслуживает студент, обнаруживший знание основного учебного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка «удовлетворительно» выставляется, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

Оценка «неудовлетворительно» выставляется студенту, обнаружившему пробелы в знаниях основного учебного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Как правило, оценка «неудовлетворительно» ставится слушателям, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании ОП без дополнительных занятий по соответствующей дисциплине.

Соответствие оценки СПбГУ и оценки ECTS (Европейской системы переноса и накопления зачётных единиц):

|  |  |  |  |
| --- | --- | --- | --- |
| Итоговый процент выполнения, % | Оценка СПбГУ при  проведении зачёта | Оценка ECTS | Оценка СПбГУ при  проведении экзамена |
| 90-100 | зачтено | A | отлично |
| 80-89 | зачтено | B | хорошо |
| 70-79 | зачтено | C | хорошо |
| 60-69 | зачтено | D | удовлетворительно |
| 50-59 | зачтено | E | удовлетворительно |
| менее 50 | не зачтено | F | неудовлетворительно |

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

Пример теста

1. Создать класс GaussNumber (h, cpp). Есть штраф за "избыточные" функции.

GaussNumber A(1, 1); // A = 1 + i

GaussNumber B = 5; // B = 5

GaussNumber C; // C = 0;

C = A; // C = 1 + i

C = 1; // C = 1;

C.add(A);// C += A

C.multiply(A); // C \*= A

2. В языке C++ (в отличии от C) невозможно скомпилировать вызов функции, объявление (или определение) которой недоступно в момент компиляции. Это связано с наличием в C++ ссылок.

Объясните в чем тут дело.

Пример недельного задания

1. Реализовать следующие функции (str.c, str.h):

- char \* strcpy(char \* destination, const char \* source);

- char \* strcat(char \* destination, const char \* source);

- int strcmp(const char \* str1, const char \* str2);

- size\_t strlen(const char \* str).

NB:

\* ключевое слово const пока можно игнорировать;

\* описание функций можно найти здесь: http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1256.pdf;

\* постарайтесь не использовать индексы для обращения к элементам массива;

\* постарайтесь реализовать функции максимально лаконично.

2. К каждой функции должна быть реализована минимум одна тестовая функция (test\_str.c, test\_str.h).

3. main и вызов тестов должны располагаться в файле test.c.

4. Требование к структуре кода:

\* сборка командой make, удаление продуктов сборки с помощью цели clean;

\* генерирование объектного файла str.o и папки bin в процессе сборки;

\* структура "проекта" должна быть как в первом задании за исключением имени исполняемого файла.

Примерный перечень теоретических вопросов для подготовки кэкзамену:

1. Шаблоны

2. Исключения

3. Последовательные контейнеры

4. Ассоциативные контейнеры

5. Алгоритмы

6. Приведение типов

7. Move семантика

8. Переменное число аргументов

9. С++11. Разное.

10. Метапрограммирование

11. Потоки и мутексы

12. Сondition variable

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

Анкета для студентов для оценки качества преподавания курса.

Просим Вас заполнить анкету-отзыв по прочитанной дисциплине. Обобщенные данные анкет будут использованы для ее совершенствования. По каждому вопросу проставьте соответствующие оценки по шкале от 1 до 10 баллов (обведите выбранный Вами балл). В  
случае необходимости впишите свои комментарии.

1. Насколько Вы удовлетворены содержанием дисциплины в  
целом?  
1 2 3 4 5 6 7 8 9 10  
Комментарий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
2. Насколько Вы удовлетворены общим стилем преподавания?  
1 2 3 4 5 6 7 8 9 10  
Комментарий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3. Как Вы оцениваете качество подготовки предложенных  
методических материалов?  
1 2 3 4 5 6 7 8 9 10  
Комментарий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Насколько Вы удовлетворены использованием  
преподавателями активных методов обучения?  
1 2 3 4 5 6 7 8 9 10  
Комментарий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Какой из модулей (разделов) дисциплины Вы считаете наиболее полезным, ценным с точки зрения дальнейшего обучения и/или  
применения в последующей практической деятельности?  
Комментарий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Что бы Вы предложили изменить в методическом и  
содержательном плане для совершенствования преподавания данной  
дисциплины?  
Комментарий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

СПАСИБО!

**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 и/или ученое звание, имеющие опыт планирования и организации учебного процесса, а также главные и ведущие специалисты в этой области.

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

Учебно-вспомогательный и инженерно-технический персонал должен иметь соответствующее высшее образование, и обладать навыками организации работы с пользовательскими программными продуктами в локальной сети компьютерного класса и в Интернете.

**3.3. Материально-техническое обеспечение**

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

Стандартно оборудованные аудитории для проведения лекционных занятий. Для проведения практических занятий - компьютерный класс на 15 рабочих мест с учетом преподавательского.

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

Универсальные офисные компьютеры средней производительности, объединенные в локальную сеть, мультимедийное оборудование (проектор, экран).

Лекционная аудитория должна иметь доску для записей, проектор.   
Для реализации программы необходим доступ преподавателей к офисной технике (компьютер, копировальный аппарат, принтер, сканер), а также достаточное количество расходных материалов к ней, выделенных для использования в учебном процессе.

**3.3.3 Характеристики специализированного оборудования**

Не требуется.

**3.3.4 Характеристики специализированного программного обеспечения**

Ubuntu Linux, make, gcc, gdb, git, svn.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

мел, фломастеры цветные, губки, бумага формата А4, канцелярские товары, картриджи принтеров, флеш-накопители и др. в объёме, необходимом для организации и проведения занятий.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы:**

1. Бьерн Страуструп, Дизайн и эволюция языка C++

2. Брайан Керниган, Деннис Ритчи, Язык программирования C

3. Конова, Е.А. Алгоритмы и программы. Язык С++ [Электронный ресурс] : учебное пособие / Е.А. Конова, Г.А. Поллак. — Электрон. дан. — Санкт-Петербург : Лань, 2018. — 384 с. — Режим доступа: https://proxy.library.spbu.ru:2190/book/103905.

**3.4.2 Список дополнительной литературы**

1. Брайан Керниган, Роб Пайк, Практика программирования.
2. Страуструп Б. Язык программирования С++ / Б. Страуструп. – СПб.: «Невский диалект», М.: Бином, 2008. – 1104 с

**3.4.3 Перечень иных информационных источников**

нет  
  
**Раздел 4. Разработчики программы**

Линский Евгений Михайлович, к.т.н., evgeny.linsky@gmail.com