

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
Московский государственный университет имени М.В. Ломоносова
Факультет биоинженерии и биоинформатики

УТВЕРЖДАЮ

Декан
факультета биоинженерии
и биоинформатики,
академик

_____/В.П. Скулачев /

« ____ » _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины:

Основы языка программирования C++

Уровень высшего образования:
специалитет

Направление подготовки (специальность):

06.05.01 Биоинженерия и биоинформатика

Форма обучения:

очная

Рабочая программа рассмотрена и одобрена

Ученым советом факультета

(протокол № _____, _____)

Москва 20__

Рабочая программа дисциплины разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по специальности 06.05.01 «Биоинженерия и биоинформатика» (программы специалитета) в редакции приказа МГУ от 30 декабря 2016 г.

Год (годы) приема на обучение – 2016, 2017, 2018, 2019.

© Факультет биоинженерии и биоинформатики МГУ имени М.В. Ломоносова

Программа не может быть использована другими подразделениями университета и другими вузами без разрешения факультета.

Цель и задачи дисциплины

Цель дисциплины – изучение студентами языка C++.

Задачи дисциплины – ознакомление с основными концепциями C++, создание программ на C++.

1. Место дисциплины в структуре ОПОП ВО – факультатив.

2. Входные требования для освоения дисциплины, предварительные условия (если есть): необходимо, чтобы предварительно были освоены дисциплины, читаемые на факультете, «Практическая информатика», «Практическая биоинформатика».

3. Планируемые результаты обучения по дисциплине:

Знать:

Основные концепции C++: классы, наследование, перегрузка функций и операторов, шаблоны, пространства имён

Уметь:

Создавать программы на C++

Владеть:

На базовом уровне стандартной библиотекой C++ и средствами для организации инфраструктуры по созданию программного проекта на C++.

Иметь опыт:

в простейшем проектировании кода в объектно ориентированной модели, в чтении и обработке данных из текстовых и бинарных файлов, специфичных для биоинформатики.

4. Формат обучения – лекционные занятия.

5. Объем дисциплины (модуля) составляет 1 з.е., в том числе 32 академических часов, отведенных на контактную работу обучающихся с преподавателем, 4 академических часа на самостоятельную работу обучающихся.

6. Краткое содержание дисциплины (аннотация):

В результате обучения по данной дисциплине студенты познакомятся в общих чертах с широко используемом в области информационных технологий языком C++. Студенты будут способны писать не сложные программы на C++, так же у студентов будет понимание, когда для обработки биоинформатических данных целесообразно использовать компилируемые в машинный код языки, в том числе C++.

Наименование и краткое содержание разделов и тем дисциплины, Форма промежуточной аттестации по дисциплине	Всего (часы)	В том числе			
		Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, часы		Самостоятельная работа обучающегося, часы (виды самостоятельной работы – эссе, реферат, контрольная работа и пр. – указываются при необходимости)	
		Занятия лекционного типа	Занятия семинарского типа	Всего	
Обзор языков программирования, место C	2	2	0	2	0

и С++ среди других языков программирования					
Инфраструктура и инструментальная среда для работы с языками программирования С и С++	5	4	0	4	1 (Домашнее задание: работа с репозиториями исходного кода)
Структура С программы	6	6	0	6	0
Стандартная библиотека языка С	3	2	0	2	1 (Домашнее задание: Работа с последовательностями)
Обзор С++, основные не объектно ориентированные отличия от С.	4	4	0	4	0
Классы, пространства имён области видимости.	2	2	0	2	0
Полиморфизм функций, наследование.	4	4	0	4	0
Перегрузка операций операторы приведения типов	4	4	0	4	0
Шаблоны в С++	3	2	0	2	1(Домашнее задание: работа с последовательностями, преобразование в С++ код.)
Стандартная библиотека С++	2	2	0	2	0
Промежуточная аттестация: зачет					1 (количество часов, отведенных на промежуточную аттестацию)
Итого	36		32		4

7. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине

7.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.

Примеры домашних заданий:

Задание поиск «генов»

Требуется написать программу с контролем ошибок. Со стандартного потока ввода вводится две последовательности. Последовательность РНК: текст в алфавите {A,a,U,u,G,g,C,c,-,пробельный символ}. Признаком конца в последовательности служит ввод символа точка. Последовательность белка. Аналогично с последовательностью РНК данная последовательность может включать пробельные символы и символ минус.

В таблице представлены правила преобразования РНК в Белок. Требуется вывести координаты начала и конца «гена» кодирующего белок в последовательности (игнорируя пробельные символы и символы минусик). Требуется вывести координаты начала и конца «гена» в тексте (номер строки, и порядковый номер в строчке). Требуется так же саму последовательность найденного «гена» вывести на экран. Находки должны быть разделены переводом строки, а из печатаемого фрагмента удалены все пробельные символы и символы минусов.

Задание «упаковка/распаковка последовательностей ДНК/РНК»

Требуется написать программу, которая последовательности ДНК и РНК, написанные как текст упаковывает в файл в бинарном формате хранения. Предполагается, что в упакованном виде будет использоваться только необходимое число бит, для кодирования алфавита {'a', 't', 'g', 'c', 'u', конец_строки}. В качестве внутреннего представления в оперативной памяти для строки предполагается использование массива int-ов.

Программа на вход принимает направление перекодирования (направление перекодирования задаётся словами «pack», «unpack») и два имени файла. Первый файл источник данных, второй приёмник. Файл с последовательностями в несжатом формате имеет так называемый FASTA формат (https://en.wikipedia.org/wiki/FASTA_format). Файл в который производится упаковка может иметь произвольный формат. Требования к этому файлу следующие:

1. В результате упаковки, его размер должен оказаться существенно меньше, чем размер файла с исходными последовательностями (это должно происходить из-за битовой упаковки данных, а не из-за удаления незначимых пробельных и прочих символов из исходного файла).
2. В результате упаковки должны сохраниться имена последовательностей.

В FASTA формате информация о последовательности начинается с символа '>', далее, до символа перевода строки следует имя последовательности. После, с новой строки следуют нуклеотиды задающие собственно саму последовательность. В тексте могут встречаться дополнительно пробельные символы, переводы строк, знаки пунктуации, цифры и т.п. Всё это нужно игнорировать и не помещать в представление последовательности в оперативной памяти. Последовательность заканчивается либо в случае конца файла, либо при встрече символа '>' в начале строки. Пример файла прилагается к заданию.

При преобразовании из сжатого в развёрнутое состояние необходимо, чтобы последовательность была отображена в человеко-читаемом виде. Для этого можно каждую группу из 10 символов окружать пробелами, а после каждой шестой такой группы вставлять перенос строки.

Примеры последовательностей можно добыть из архива ENA на сайте (<http://www.embl.org>).

7.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.

Пример контрольной работы язык Си:

1. Напишите функцию, преобразующие бинарные файлы с записями целого типа (uint32_t) от little endian к big endian. Запрещается в памяти хранить более 2-х байт из файла. (В little endian — в байте ближнем к началу файла содержатся старшие разряды числа, В big endian - в байте ближнем к началу файла содержатся младшие разряды числа). Файл разрешается открывать не более одного раза.
2. Дан двунаправленный кольцевой список, требуется написать функцию, удаляющую из списка вправо все строки лексикографически меньшие указанной в параметре. Начало кольца переместить в позицию первого лексикографического элемента большего либо равному параметру функции. Если кольцо оказывается пустым или было пустым, то параметр помещается в кольцо.
3. Напишите функцию осуществляющую в строке замену вхождений слова указанного в параметре функции на конструкцию **--:число:--**, где число задаёт номер вхождения слова в исходную строку.
4. Написать программу. Из файла извлекаются 2 битовых массива. В аргументах main передаются смещения начал массивов в файле (в байтах), число бит (задающих размер массива), имя файла. На стандартный поток вывода распечатывается число отличающихся бит в массивах.

Пример контрольной работы C++:

1. Есть ли ошибки в реализации функций `B::g()` и `main()`? Если есть, **объясните**, в чем они заключаются. Для всех правильных операторов этих функций укажите, из какой области видимости выбираются участвующие в их записи имена, используя операцию разрешения области видимости «`::`».

```
int x = 0;
void f (int a, int b){x = a+b;}
class A {
int x;
public:
void f () {x = 2;}
};
class B: public A {
public:
void f (int a){::x = a;}
void g ();
};
void B::g() {
f();
f(1);
f(5 , 1);
x = 2;
}
int main () {
B b;
f(5);
f('+', 6);
return 0;
}
```

2. Что будет выдано в стандартный канал вывода при работе следующей программы?

```
class X;
void F(X & x, int n);
class X {
public: X() { try { F(*this, -2); cout << 1 << endl; }
catch (X) { cout << 2 << endl; }
catch (int) { cout << 3 << endl; } }
X (X &) { cout << 12 << endl; }
};
class Y: public X {
public: Y () {cout << 4 << endl;}
Y (Y & a) {cout << 5 << endl;}
~Y () {cout << 6 << endl;}
};
void F(X & x, int n) { try { if (n < 0) throw x;
if (n > 10) throw 1;
cout << 7 << endl; }
catch (int) { cout << 8 << endl; }
catch (X&) { cout << 9 << endl; throw; }
}
int main() { try { Y a; }
catch (...) { cout << 10 << endl; }
cout << 11 << endl;
}
```

3. Для класса `rational` опишите перегруженную операцию инкремента (префиксный и постфиксный варианты) так, чтобы были верными все действия функции `g()`. Семантика этой операции – увеличение на 1 значений полей `p` и `q`. Возможно ли это? Если нет – объясните почему; если можно – опишите и объясните, какие действия будут выполняться при вызове функции `g()`?

```
class rational { int p, q;
```

```

        public: rational (int a , int b ) { p = a; q = b;} //...
    };

void g()
{ rational x(1,2), z(1,1);    const rational y(1,8);
  z = x++;
  z = y++;
  z = ++x;
  z = ++y; }

```

Шкала и критерии оценивания результатов обучения по дисциплине.

Результаты обучения	«Неудовлетворительно»	«Удовлетворительно»	«Хорошо»	«Отлично»
Знания: Основных концепций C++: классы, наследование, перегрузка функций и операторов, шаблоны, пространства имён	Знания отсутствуют	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
Умения: Создавать программы на C++	Умения отсутствуют	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности неприципиального характера)	Успешное и систематическое умение
Владения: На базовом уровне стандартной библиотекой C++ и средствами для организации инфраструктуры по созданию программного проекта на C++	Навыки владения отсутствуют	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач

8. Ресурсное обеспечение:

Перечень основной и дополнительной литературы

1. Брайан Керниган, Деннис Ритчи. Язык программирования C. — Москва: Вильямс, 2015. — 304 с.
2. А.В.Столяров. Операционная среда ОС UNIX для изучающих программирование. (Архитектура ЭВМ и СПО. Пособие для выполнения лабораторных работ в ОС UNIX. М.: МГТУГА, 2008)
3. Курячий Г. В., Маслинский К. А. Операционная система Linux. — ALT Linux Москва, 2010. — С. 347.

4. Скотт Чако́н, Бен Штрауб Git для профессионального программиста – Питер, Санкт-Петербург, 2016, 496 стр.
5. Бьерн Страуструп Дизайн и эволюция языка C++ – ДМК Пресс, Санкт-Петербург, 2006, 448 стр.
6. Бьерн Страуструп: Язык программирования C++. Специальное издание - Бином, 2016, 1136 стр.
7. Андрей Александреску Современное проектирование на C++: Обобщенное программирование и прикладные шаблоны проектирования, Вильямс, 2015, 336 стр.
8. Николай М. Джосаттис Стандартная библиотека C++: справочное руководство - Вильямс, 2016, 1136 стр.
9. Скотт Мейерс Эффективный и современный C++. 42 рекомендации по использованию C++11 и C++14 - Вильямс, 2016
10. Скотт Мейерс Эффективное использование STL - Питер, 2002 , 224 стр.

Перечень лицензионного программного обеспечения (при необходимости)

Специальное лицензионное программное обеспечение не требуется.

Перечень профессиональных баз данных и информационных справочных систем

Требуется доступ к специализированным базам данных биологических последовательностей, таких как EnsEMBL, UniProt. Требуется доступ к сервису поиска нуклеотидных и аминокислотных последовательностей blast.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости)

1. Сайт со справочной информацией о стандарте C++ - <http://cppreference.com>
2. Сайт online сервиса поиска последовательностей: <https://blast.ncbi.nlm.nih.gov/Blast.cgi>
3. Сайт геномных сборок: <http://www.ensembl.org/index.html>

Описание материально-технического обеспечения.

Для проведения дисциплины требуется компьютерный класс, для приёма домашних заданий, проведения семинаров и закрепления студентами демонстрируемого на лекции материала, в том числе модификации и создания программного кода. В классе требуется доска и проектор для демонстрации кода.

Необходим сервер репозитория исходного кода, а так же web сервер, для публикации заданий и результатов проверок выполнения заданий и результатов проверок контрольных работ.

У каждого студента на сервере репозитория должен быть свой персональный репозиторий, доступный только студенту и преподавателям, ведущим занятия, а так-же собственная учётная запись и домашний каталог на машинах в компьютерном классе.

Предпочтительна программное окружение UNIX-подобных операционных систем.