

**Санкт-Петербургский филиал федерального государственного
автономного образовательного учреждения высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет Санкт-Петербургская школа
физико-математических и компьютерных наук
Департамент информатики

Рабочая программа дисциплины
Алгоритмические основы программирования
для майнора «Основы программирования»
уровень бакалавриат

Разработчик: Дворкин Михаил Эдуардович, mdvorkin@hse.ru

Утверждена Академическим руководителем образовательной программы
«31» августа 2018 г.

А.В. Омельченко _____

Санкт-Петербург, 2018

Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.

1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает требования к образовательным результатам и результатам обучения студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих дисциплину «Алгоритмические основы программирования», учебных ассистентов и студентов направления 01.03.02 «Прикладная математика и информатика» подготовки бакалавра, обучающихся по бакалаврской программе «Прикладная математика и информатика» и изучающих дисциплину «Алгоритмические основы программирования».

Рабочая программа дисциплины разработана в соответствии с:

- Образовательным стандартом НИУ ВШЭ по направлению подготовки 01.03.02 «Прикладная математика и информатика» (уровень бакалавриата), утвержденным ученым советом Национального исследовательского университета «Высшая школа экономики», протокол от 03.03.2017 №02.
- Образовательной программой «Прикладная математика и информатика» направления подготовки 01.03.02 «Прикладная математика и информатика»;
- Объединенным учебным планом университета по образовательной программе «Прикладная математика и информатика», утвержденным в 2018 г.

2. Цели освоения дисциплины

Целями освоения дисциплины «Алгоритмические основы программирования» являются формирование у студентов теоретических знаний и практических навыков по основам методологии программирования.

Целью дисциплины - сформировать у студентов базовое представление об алгоритмах и структурах данных, применяющихся в современном программировании, а также о возможностях реализации алгоритмов и структур данных на скриптовых языках программирования..

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ФГОС	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
Способность учиться, приобретать новые знания, умения, в том числе в области, отличной от профессиональной	УК-1	РБ СД МЦ	Знает основные способы познания, их эволюцию Умеет применять знания основных предметных областей в рамках профессиональной деятельности Имеет представление об основных методах и способах получения новых знаний	Лекции, подготовка к практическим занятиям, работа на практических занятиях, индивидуальная самостоятельная работа	Домашние задания, письменный зачет, письменный экзамен

<p>Способен вести исследовательскую деятельность, включая анализ проблем, постановку целей и задач, выделение объекта и предмета исследования, выбор способа и методов исследования, а также оценку его качества</p>	<p>УК-6</p>	<p>РБ СД МЦ</p>	<p>Знает различные методы поиска информации Умеет использовать информационно-компьютерные системы для поиска и обработки информации при мобильной разработке Владеет навыками и опытом поиска и обработки информации, необходимой при ИТ разработке</p>	<p>Лекции, подготовка к практическим занятиям, работа на практических занятиях, индивидуальная самостоятельная работа</p>	<p>Домашние задания, письменный зачет, письменный экзамен</p>
<p>Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач</p>	<p>ОПК-2</p>	<p>РБ СД МЦ</p>	<p>Знание основных методологий программирования Умение формализовать прикладную задачу и выбрать соответствующую методологию программированию, наиболее подходящую для решения данной задачи Владение навыками алгоритмизации и реализации алгоритмов на языках программирования с использованием различных методологий программирования</p>	<p>Лекции, подготовка к практическим занятиям, работа на практических занятиях, индивидуальная самостоятельная работа</p>	<p>Домашние задания, письменный зачет, письменный экзамен</p>
<p>Способен применять и модифицировать математические модели для решения задач в области профессиональной деятельности</p>	<p>ОПК-3</p>	<p>РБ СД МЦ</p>	<p>Знание основных библиотек языка Python Умение использовать данные библиотеки при решении прикладных задач Владение навыками разработки собственных библиотек, необходимых для решения различных прикладных задач</p>	<p>Лекции, подготовка к практическим занятиям, работа на практических занятиях, индивидуальная самостоятельная работа</p>	<p>Домашние задания, письменный зачет, письменный экзамен</p>

Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-4	РБ	Знание основных принципов программирования на языке Python Умение создавать приложения с использованием языка Python Владение методами отладки, редактирования и разработки программ на языке Python	Лекции, подготовка к практическим занятиям, работа на практических занятиях, индивидуальная самостоятельная работа	Домашние задания, письменный зачет, письменный экзамен
		СД			

4. Место дисциплины в структуре образовательной программы.

Для программы «Прикладная математика и информатика» направления подготовки 01.03.02 «Прикладная математика и информатика» настоящая дисциплина относится к циклу дисциплин майнор.

Курс рассчитан на 60 часов аудиторной нагрузки, из них 40 часов лекций, общим объемом 5 зачетных единиц (190 часов).

5. Тематический план учебной дисциплины.

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Основы программирования	50	10	5	-	35
2	Основные понятия теории алгоритмов	40	10	5	-	25
3	Основные структуры данных	50	10	5	-	35
4	Сортировки	50	10	5	-	35
ИТОГО		190	40	20	-	130

6. Содержание дисциплины

Раздел 1
Основы программирования

Тема 1	Синтаксис языка Python Целые числа, ввод-вывод, строки; условный оператор; цикл while; вещественные числа и операции над ними. функция и рекурсия; кортежи и списки; цикл for; множества и словари
Тема 2	Базовые понятия объектно-ориентируемого программирования
<u>Раздел 2</u> Основные понятия теории алгоритмов	
Тема 1	Сложность алгоритмов. O-символика
Тема 2	Алгоритмы над массивами Динамическое программирование. Поиск. Метод «разделяй и властвуй»
<u>Раздел 3</u> Основные структуры данных	
Тема 1	Массивы и списки
Тема 2	Вектор, стек, очередь, дек. Куча
<u>Раздел 4</u> Сортировки	
Тема 1	Квадратичные алгоритмы
Тема 2	Сортировки работающие за NLogN Сортировка кучей; сортировка слиянием; быстрая сортировка

7. Оценочные средства для текущего контроля и аттестации студента

7.1. Формы контроля знаний студентов.

Тип контроля	Форма контроля	2 курс		Параметры
		1 модуль	2 модуль	
Текущий	Домашнее задание №1	*		Письменное домашнее задание
	Домашнее задание №2		*	Письменное домашнее задание
Итоговый	Письменный экзамен		*	Экзамен в письменной форме

7.2. КРИТЕРИИ ОЦЕНКИ ЗНАНИЙ, НАВЫКОВ

Текущий контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств.

ДОМАШНЕЕ ЗАДАНИЕ №1

Домашнее задание №1 выдается студентам в одном варианте и состоит из 5 задач. Каждой задаче присвоен свой балл. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимся домашнего задания - файл со скриптом.

Задача 1. [2 балла] Числа Фибоначчи

Как известно, числа Фибоначчи определяются следующим образом:

$$F_0 = 0, F_1 = 1, F_{n+2} = F_{n+1} + F_n$$

Вам нужно написать скрипт, который считывает из stdin число N и выводит в stdout число FN .

Задача 2. [2 балла] OpenCV

Во всех .cpp и .hpp-файлах в текущей директории (с учетом всех поддиректорий рекурсивно) необходимо сгруппировать все инклюды заголовочных файлов OpenCV (<opencv2/...>) в начале файла и вставить до них строку #pragma warning(push, 0), а после — #pragma warning(pop).

Задача 3. [2 балла] BFS

Написать скрипт, который выводит в stdout все файлы текущей директории в порядке обхода в ширину.

Файлы, находящиеся на одной глубине, должны выводиться через перевод строки в лексикографическом порядке; уровни глубины должны разделяться пустыми строками, допускается пустая строка в конце вывода.

Задача 4. [2 балла] Украшаем find

Написать скрипт, который оформляет вывод команды find в древо-видную структуру.

Пример входной директории

```
$ find
```

```
.
./answers
./answers/Tree.hs
./answers/bfs.sh
./hw
./hw/Makefile
./hw/unix_hw2.tex
```

Пример запуска решения

```
$ ./4.sh
```

```
.
|-answers
|---Tree.hs
|---bfs.sh
|-hw
|---Makefile
|---unix_hw2.tex
```

Задача 5. [2 балла] Стек

Написать скрипт, реализующий стек (см. пример).

Пример работы скрипта

```
$ ./stack.sh push "Line 1" && echo "OK"
```

```
OK
```

```
$ ./stack.sh push "Line 2" && echo "OK"
```

```
OK
```

```
$ ./stack.sh pop && echo "OK"
```

Line 2

OK

```
$ ./stack.sh pop && echo "OK"
```

Line 1

OK

```
$ ./stack.sh pop || echo "FAIL"
```

FAIL

Ограничения

Если потребуются временные файлы, размещать их можно только в скрытой папке `.au_cw_stack/` в домашней директории текущего пользователя (создавать ее нужно самостоятельно внутри скрипта).

Ограничения

$\$1 \geq 2$.

Критерии оценивания и шкала оценки домашнего задания №1.

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Решено задач на 8 или более баллов
«Хорошо» (6-7)	Решено задач на 6-7 баллов
«Удовлетворительно» (4-5)	Решено задач на 4-5 баллов
«Неудовлетворительно» (0-2)	Решено задач на менее чем 4 балла

ДОМАШНЕЕ ЗАДАНИЕ №2

Домашнее задание №2 выдается студентам в одном варианте. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Требуется реализовать программу умножающую квадратные матрицы целых чисел используя алгоритм Штрассена (или его модификацию). Скрипт должен читать две матрицы из стандартного потока ввода (для чтения можно использовать функцию `input` или `numpry.loadtxt`).

Первой строкой на вход скрипту подается целое число - размер перемножаемых квадратных матриц, далее следуют $2 * n$ строк описывающих строки матриц. Числа в строках разделены пробелами.

Ваша программа должна выдать результирующую матрицу на стандартный поток вывода (можете использовать `print`), числа строки результирующей матрицы должны быть разделены ненулевым количеством пробелов или табуляций, и каждая строка матрицы должна выводиться на новой строке.

Пример ввода:

3

1 0 0

0 1 0
0 0 1
2 0 0
0 2 0
0 0 2

Пример вывода:

2 0 0
0 2 0
0 0 2

Критерии оценивания и шкала оценки домашнего задания №4.

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Написанный скрипт работает корректно и оптимально
«Неудовлетворительно» (0-2)	Скрипт не соответствует требованиям

ИТОГОВЫЙ КОНТРОЛЬ

Проверка качества освоения дисциплины производится в форме письменного экзамена.

ПИСЬМЕННЫЙ ЭКЗАМЕН

Письменный экзамен проводится в форме ответов на вопросы экзаменационного билета. Экзаменационный билет представляет собой один из алгоритмов, которые надо было реализовать в домашнем задании. На подготовку ответа выделяется 2,5 часа.

Критерии оценивания и шкала оценки устного экзамена

Оценка	Критерии выставления оценки
«Отлично» (8-10)	Дан развернутый ответ на поставленный вопрос. Материал изложен последовательно. Имеются логичные и аргументированные выводы.
«Хорошо» (6-7)	Дан развернутый ответ на поставленный вопрос. Материал изложен непоследовательно. Имеются логичные и аргументированные выводы.
«Удовлетворительно» (4-5)	Ответ на вопрос не является полным. Материал изложен непоследовательно. Отсутствуют выводы.
«Неудовлетворительно» (0-2)	Ответ на вопрос является неверным. Материал изложен непоследовательно. Отсутствуют выводы.

7.3 Порядок формирования оценок по дисциплине

Преподаватель учитывает оценку за текущий контроль (домашние задания).

$$O_{\text{накопленная}} = 0,5 * O_{\text{д/з1}} + 0,5 * O_{\text{д/з2}}$$

Действует способ округления накопленной оценки текущего контроля в пользу студента (при значениях от 0,1 до 0,4 оценка округляется в меньшую сторону, от 0,5 до 0,9 – в большую)

Результирующая оценка за дисциплину рассчитывается следующим образом:

$$O_{\text{Результирующая}} = 0,5 O_{\text{накопленная}} + 0,5 O_{\text{экзамен}}$$

На экзамене студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Основная литература

1. Черпаков, И. В. Основы программирования : учебник и практикум для прикладного бакалавриата / И. В. Черпаков. — М. : Издательство Юрайт, 2018. — 219 с.
2. Fundamentals of Python®: Data Structures / By: Kenneth A. Lambert. Course Technology. 2013

8.2. Дополнительная литература

1. Федоров, Д. Ю. Программирование на языке высокого уровня python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — М. : Издательство Юрайт, 2018. — 126 с.
2. Операционные системы, среды и оболочки: Учебное пособие / Т.Л. Партыка, И.И. Попов. - 5-е изд., перераб. и доп. - М.: Форум: НИЦ ИНФРА-М, 2013. - 560 с.
3. Python Algorithms: Mastering Basic Algorithms in the Python Language; Second Edition / Magnus Lie Hetland. Apress. 2014

9. Перечень учебно-методического обеспечения для самостоятельной работы, обучающихся по дисциплине.

Самостоятельная работа может рассматриваться как организационная форма обучения - система педагогических условий, обеспечивающих управление учебной деятельностью по освоению знаний и умений в области учебной деятельности без посторонней помощи. Студенту нужно четко понимать, что самостоятельная работа – не просто обязательное, а необходимое условие для получения знаний по дисциплине и развитию компетенций, необходимых в будущей профессиональной деятельности.

Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных на лекциях теоретических знаний;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- формирования практических (общеучебных и профессиональных) умений и навыков;
- развития исследовательских умений;

- получения навыков эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности.

В учебном процессе выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа - планируемая учебная работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная программой учебной дисциплины, раскрывающей и конкретизирующей ее содержание, осуществляется студентом инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует источники для работы, характеризует наиболее рациональную методику самостоятельной работы, демонстрирует ранее выполненные студентами работы и т. п.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать индивидуальные особенности студента.

Самостоятельная работа может осуществляться индивидуально или группами студентов online и на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности.

Контроль результатов внеаудиторной самостоятельной работы осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине на семинарских или контрольных занятиях.

Для представления результатов самостоятельной работы рекомендуется:

Составить план выступления, в котором отразить тему, самостоятельный характер проделанной работы, главные выводы и/или предложения, их краткое обоснование и практическое значение – с тем, чтобы в течение 3 – 5 минут представить достоинства выполненного самостоятельно задания.

Подготовить иллюстративный материал виде презентации для использования во время представления результатов самостоятельной работы в аудитории. Конкретный вариант наглядного представления результатов определяется форматом аудиторного занятия и задания преподавателя.

10. Материально-техническое обеспечение дисциплины и информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения информационных справочных систем (при необходимости).

Для проведения всех занятий используется проектор и компьютер для проекции слайдов.