

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**  
**Проректор по учебной работе**

**А.А. Воронов**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Основы программирования на Python
<b>по направлению:</b>	Биотехнология
<b>профиль подготовки:</b>	Биотехнология
	Физтех-школа Биологической и Медицинской Физики кафедра информатики и вычислительной математики
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 0 час.

семинары: 0 час.

лабораторные занятия: 60 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Количество контрольных работ, заданий: 1

Программу составили:

Т.Ф. Хирьянов, старший преподаватель

М.Н. Герцев, канд. физ.-мат. наук

Программа обсуждена на заседании кафедры информатики и вычислительной математики 06.02.2020

## Аннотация

В настоящем курсе будут даны основы алгоритмического языка программирования Python 3, принципы создания качественного читабельного кода. Студенты изучат приёмы разработки программ «сверху-вниз» и «снизу-вверх», идеологию модульного и объектно-ориентированного подхода. Обучающиеся разберут типовые решения, применяемые для создания программ.

### 1. Цели и задачи

#### Цель дисциплины

научить студентов программировать простые графические приложения на языке Python 3 как самостоятельно, так и в группе, с использованием системы контроля версий git и соблюдением принципов качества кода.

#### Задачи дисциплины

- формирование у обучающихся базовых знаний о синтаксисе языка Python 3 и его возможностях;
- формирование культуры создания читабельного кода;
- формирование умения осуществлять декомпозицию проекта ПО на функции, объекты и модули;
- формирование навыка проектирования и разработки ПО с использованием системы контроля версий, в том числе в рабочей группе.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основы алгоритмического языка программирования Python 3;
- принципы создания качественного читабельного кода;
- приёмы разработки программ «сверху-вниз» и «снизу-вверх»;
- идеологию модульного и объектно-ориентированного подхода;
- типовые решения, применяемые для создания программ.

уметь:

- разрабатывать читабельные программы на языке программирования Python 3;
- использовать как встроенную, так и доступную в Сети документацию по библиотекам Python 3;
- подключать дополнительные модули и стандартные модули и пакеты Python 3;
- создавать дополнительные модули и пакеты на Python 3 для основной программы;
- применять объектно-ориентированный подход для написания программ;
- разрабатывать программы как индивидуально, так и в команде, с использованием современных средств написания и отладки программ.

владеть:

- одной из интегрированных сред разработки программ для языка Python 3;
- интерактивной консолью Python 3 для простых вычислений;
- основными командами системы контроля версий git;
- основным инструментарием библиотеки Tkinter;
- средствами отладки и интроспекции на языке Python 3.

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Синтаксис языка Python 3			8	8
2	Проектирование ПО			8	2
3	Структурное программирование			4	4
4	Модульное программирование			4	2
5	Объектно-ориентированное программирование			4	8
6	Групповая разработка программ			6	4
7	Событийно-ориентированное программирование			6	4
8	Разработка игры «Сапёр»			8	11
9	Семестровый проект			12	32
Итого часов				60	75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

##### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

###### Семестр: 1 (Осенний)

###### 1. Синтаксис языка Python 3

Интерактивный режим Python 3 и интегрированные среды разработки.

Условный оператор if. Вложенные и каскадные ветвления.

Логические операции and, or, not.

Циклы while и for. Инструкции управления циклом.

Генератор прогрессий range().

Вложенные циклы.

Описание функций без параметров и с параметрами.

Кортежи переменных. Обмен значений.

Итерируемые объекты и цикл for.

Золотой фонд Python: коллекции tuple, list, set, dict.

Изменяемость списка и операции с ним.

Неизменяемость кортежа и операции с ним.

Список кортежей. Цикл for для двух переменных.

Разворачивание итерируемого объекта в параметры функции.

Генераторы списков, кортежей, множеств.

Ссылочная модель данных. Присваивания в Python.

Интроспекция. Оператор is.

Специфика Python: duck typing.

Значения параметров по умолчанию.

Именованные параметры функций.

## 2. Проектирование ПО

Проработка интерфейсов функций.

Рефакторинг.

Введение в ООП проектирование.

Проработка интерфейсов, контрактов и ответственности классов.

## 3. Структурное программирование

Инкапсуляция ответственности в функцию.

Проектирование «снизу-вверх».

Декомпозиция.

Проектирование «сверху-вниз».

## 4. Модульное программирование

Цель и принцип разделения на модули.

Создание модулей и пакетов.

Возможности инструкции `import`.

Проработка и документация интерфейса модуля.

Локализация переменных.

## 5. Объектно-ориентированное программирование

Классы и объекты в Python.

Создание и инициализация объекта.

Инкапсуляция ответственности в класс.

Принцип единственной ответственности класса.

Отношения между классами: наследование, композиция, ассоциация.

Диаграмма классов UML.

## 6. Групповая разработка программ

Каскадная модель разработки Waterfall.

Итеративная разработка.

Распределение ролей в проекте.

Документация проекта.

Необходимость контроля версий. Терминология.

Система контроля версий `git`

Создание и клонирование репозитория: `git init`, `git clone`, `git status`.

Контроль изменений: `git diff`, `git add`, `git commit`, `git log`, `git blame`.

Ветки `git`: `git branch`, `checkout`, `merge`.

Система отслеживания ошибок в проекте и управления проектом.

Взаимная вычитка кода и `approve`.

## 7. Событийно-ориентированное программирование

Событийная модель построения приложения.

Виджеты, события и обработчики событий.

Свойства и упаковка виджетов.

Основы библиотеки Tkinter

Создание интерактивной графической программы.

Виджеты Tkinter и их упаковка в главное окно программы.

Tkinter Canvas: методы, идентификаторы и теги.

Переменные с обратной связью в tkinter.

#### 8. Разработка игры «Сапёр»

Разработка интерфейса игры с двумерной сеткой. Построение программы с разбиением на модули. Выделение объектов. Практика обработки ввода пользователя. Вычитка кода и исправление ошибок.

#### 9. Семестровый проект

Разработка архитектуры программного продукта.

Разработка плана создания ПО.

Распределение ролей участников проекта.

Взаимодействие через GitHub.

Коворкинг. Обсуждение с преподавателем и ментором.

Сдача проекта.

### **5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Аудитории для лабораторных работ должны быть оборудованы компьютерами по числу обучающихся с операционной системой GNU/Linux и интерпретатором Python 3, подключенными к сети Интернет.

Для работы преподавателя лабораторных работ нужна также меловая или маркерная доска, а также мультимедиа проектор или большой экран с возможностью подключения компьютера преподавателя через D-SUB или HDMI.

### **6.Перечень рекомендуемой литературы**

#### Основная литература

1. Программирование на Python 3 : Подробное руководство [Текст] = Programming in Python 3 : [учеб. пособие для вузов] / М. Саммерфилд; пер. с англ. А. Киселева .— СПб : Символ-Плюс, 2015 .— 608 с.
2. Программирование на Python 3, подробное руководство/М. Саммерфилд,-СПб, Символ-Плюс, 2020

#### Дополнительная литература

1. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] : [учеб. пособие для вузов] / Э. Гамма [и др.] ; [пер. с англ. А. Слинкина] .— СПб. : Питер, 2012 .— 368 с.

### **7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

Не используются

### **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

Для командной работы над проектами под системой контроля версий git обучающимся во время занятий необходим доступ в Интернет, в частности к сайту <https://github.com>.

Интерпретатор Python 3..

Среды разработки IDLE, PyCharm Community Edition, IDE Spyder, Microsoft Visual Studio Code.

### **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Для освоения дисциплины студент должен выполнять и сдавать преподавателю лабораторные работы в срок.

Тексты работ расположены на сайте курса, а их выполнение предполагает загрузку кода проектов в git-репозиторий.

Если вовремя выполнения лабораторной работы возникают вопросы, следует обращаться к преподавателю или ментору.

Последние несколько недель в семестре отводятся для создания семестровой или курсовой работы, которая обязательна к выполнению.

Курсовая выполняется группой из двух-трёх студентов, поэтому важно распределить ответственность за компоненты кода между студентами, организовать согласованную совместную работу.

Обучающимся следует учесть, что система git позволяет преподавателю отслеживать равномерность выполнения курсовой работы в течение срока её подготовки.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

**по направлению:** Биотехнология  
**профиль подготовки:** Биотехнология  
Физтех-школа Биологической и Медицинской Физики  
кафедра информатики и вычислительной математики  
**курс:** 1  
**квалификация:** бакалавр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

**Разработчики:**

Т.Ф. Хирьянов, старший преподаватель  
М.Н. Герцев, канд. физ.-мат. наук

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Основы программирования на Python» обучающийся должен:

### знать:

- основы алгоритмического языка программирования Python 3;
- принципы создания качественного читабельного кода;
- приёмы разработки программ «сверху-вниз» и «снизу-вверх»;
- идеологию модульного и объектно-ориентированного подхода;
- типовые решения, применяемые для создания программ.

### уметь:

- разрабатывать читабельные программы на языке программирования Python 3;
- использовать как встроенную, так и доступную в Сети документацию по библиотекам Python 3;
- подключать дополнительные модули и стандартные модули и пакеты Python 3;
- создавать дополнительные модули и пакеты на Python 3 для основной программы;
- применять объектно-ориентированный подход для написания программ;
- разрабатывать программы как индивидуально, так и в команде, с использованием современных средств написания и отладки программ.

### владеть:

- одной из интегрированных сред разработки программ для языка Python 3;
- интерактивной консолью Python 3 для простых вычислений;
- основными командами системы контроля версий git;
- основным инструментарием библиотеки Tkinter;
- средствами отладки и интроспекции на языке Python 3.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Контроль освоения обучающимися учебного материала осуществляется в виде устного опроса в начале занятия по теме прошлого занятия.

## 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Самостоятельное создание за короткое время в присутствии преподавателя небольших программ на Python 3.
2. Самостоятельное или групповое создание большой программы на Python 3 в публичном репозитории в течение двух-трёх недель с последующей защитой проекта.

Перечень типовых заданий на дифференцированном зачете:

1. Реализуйте свой класс Complex для комплексных чисел
  - Добавьте конструктор класса
  - Реализуйте операции проверки на равенство, сложения, вычитания, произведения и деления комплексных чисел (eq, add, sub, mul, truediv)
  - Реализуйте операцию модуля (abs)



- Класс должен давать осмысленный вывод как при print, так и просто при вызове в ячейке ноутбука
- 2. Напишите генератор, выводящий первые n чисел Фибоначчи.
- 3. Реализуйте класс BinTree двоичного дерева, и его итератор, который обходит дерево в порядке обхода в глубину.
- 4. Решите без использования циклов средствами NumPy (каждый пункт решается в 1-2 строчки)
  - Создайте вектор с элементами от 12 до 42
  - Создайте вектор из нулей длины 12, но его пятый элемент должен быть равен 1
  - Создайте матрицу (3, 3), заполненную от 0 до 8
  - Найдите все положительные числа в np.array([1,2,0,0,4,0])
  - Умножьте матрицу размерности (5, 3) на (3, 2)
  - Создайте матрицу (10, 10) так, чтобы на границе были 0, а внутри 1
  - Создайте случайный вектор и отсортируйте его

#### Критерии оценивания

- 10 - Обучающийся ответил на все вопросы, но не с первой попытки.
- 9 - Обучающийся допустил не более одной ошибки или воспользовался помощью преподавателя.
- 8 - Обучающийся, работая самостоятельно, допустил не более двух численных ошибок в лабораторной работе.
- 7 - Обучающийся если он ответил на подавляющее большинство вопросов в лабораторной работе, может быть с помощью преподавателя или товарищей.
- 6 - Обучающийся ответил на подавляющее большинство вопросов в лабораторной работе, может быть с помощью преподавателя или товарищей.
- 5 - Обучающийся ответил на основные вопросы в лабораторной работе, может быть с помощью преподавателя или товарищей.
- 4 - Обучающийся ответил на основные вопросы в лабораторной работе
- 3 - Обучающийся ответил на некоторые вопросы в лабораторной работе.
- 2 - Обучающийся не справился с работой.
- 1 - Обучающийся демонстрирует полное отсутствие знаний по предмету или пытался выдать чужую работу за свою.

#### **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Итоговая аттестация по дисциплине осуществляется в форме дифференцированного зачета. Оценка за дифференцированный зачет выставляется с учётом оценок лабораторных работ, выполняемых в течение семестра.