

Заболотских Е.Д. Программно-методическое обеспечение освоения языка Python в учебном курсе «Технологии программирования» / ПГГПУ, Пермь, 2017. – 40с.. + Прил. [Электр. ресурс].

Рассматривается возможность создания электронного учебного модуля для изучения языка Python. Приведено описание высокоуровневых языков программирования и раскрыты их характеристики. Проанализированы имеющиеся образовательные курсы по языку Python. Описаны особенности языка Python и область его применения. Разработан цифровой учебный модуль «Язык Python» для студентов изучающих программирование.

Оглавление

Введение.....	4
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ И ОБЗОР УЧЕБНЫХ КУРСОВ ПО ПРОГРАММИРОВАНИЮ.....	6
1.1. Современные языки программирования высокого уровня.....	6
1.2. Обзор образовательных курсов по программированию на языке Python.....	15
ГЛАВА 2. РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА «ЯЗЫК PYTHON».....	21
2.1. Общая характеристика учебного модуля «Язык Python».....	21
2.2. Исходный код программы «Язык Python» и его описание.....	29
ЗАКЛЮЧЕНИЕ.....	38
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	39

Введение

В связи с наблюдаемым в настоящее время стремительным развитием персональной вычислительной техники, происходит постепенное изменение требований, предъявляемых к языкам программирования. Все большую роль начинают играть интерпретируемые языки, поскольку возрастающая мощность персональных компьютеров начинает обеспечивать достаточную скорость выполнения интерпретируемых программ. А единственным существенным преимуществом компилируемых языков программирования является создаваемый ими высокоскоростной код. Когда скорость выполнения программы не является критичной величиной, наиболее правильным выбором будет интерпретируемый язык, как более простой и гибкий инструмент программирования. В связи с этим, определенный интерес представляет рассмотрение сравнительно нового языка программирования Python, который был создан его автором Гвидо ван-Россумом в начале 90-х годов.

Актуальность выбранной темы состоит в необходимости использования в учебном образовательном процессе цифровых образовательных ресурсов для поддержки самостоятельной работы студентов.

Объект исследования: разработка цифровых образовательных ресурсов с применением современных языков и технологий программирования.

Предмет исследования: применение языка Python в разработке цифровых учебных материалов.

Цель работы: разработка программно-методического обеспечения освоения языка Python в учебном курсе «Технологии программирования» средствами инструментальной среды IDLE.

Задачи:

1. Рассмотреть современные языки программирования высокого уровня.
2. Проанализировать образовательные курсы по программированию на языке Python.

3. Выделить дидактическую структуру учебного модуля «Язык Python».
4. Программно реализовать учебный модуль «Язык Python».

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ И ОБЗОР УЧЕБНЫХ КУРСОВ ПО ПРОГРАММИРОВАНИЮ

1.1. Современные языки программирования высокого уровня

Высокоуровневый язык программирования — язык программирования, разработанный для быстроты и удобства использования программистом. Основная черта высокоуровневых языков — это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.

Так, высокоуровневые языки стремятся не только облегчить решение сложных программных задач, но и упростить портирование программного обеспечения. Использование разнообразных трансляторов и интерпретаторов обеспечивает связь программ, написанных при помощи языков высокого уровня, с различными операционными системами и оборудованием, в то время как их исходный код остаётся, в идеале, неизменным.

Такого рода оторванность высокоуровневых языков от аппаратной реализации компьютера помимо множества плюсов имеет и минусы. В частности, она не позволяет создавать простые и точные инструкции к используемому оборудованию. Программы, написанные на языках высокого уровня, проще для понимания программистом, но менее эффективны, чем их аналоги, создаваемые при помощи низкоуровневых языков. Одним из следствий этого стало добавление поддержки того или иного языка низкого уровня (язык ассемблера) в ряд современных профессиональных высокоуровневых языков программирования [10].

Python



Рис.1. Логотип Python

Python – это один из наиболее популярных современных языков программирования(рис. 1). Он пригоден для решения разнообразных задач и предлагает те же возможности, что и другие языки программирования: динамичность, поддержку ООП и кросс-платформенность. Разработку Python начал Гвидо Ван Россум (GuidoVanRossum) еще в середине 1990-х годов, поэтому к настоящему времени удалось избавиться от стандартных «детских» болезней, существенно развить лучшие стороны языка и привлечь множество программистов, использующих Python для реализации своих проектов.

Многие программисты считают, что необходимо изучать только «классические» языки программирования, такие как Java или C++, так как другие языки все равно не смогут обеспечить таких же возможностей. Однако в последнее время возникло убеждение, что программисту желательно знать более одного языка, так как это расширяет его кругозор, позволяя более творчески решать поставленные задачи и повышая его конкурентоспособность на рынке труда.

Изучить в совершенстве два таких языка как Java и C++ достаточно сложно и заняло бы много времени; кроме того, многие аспекты этих языков противоречат друг другу. В то же время Python идеально подходит на роль второго языка, так как он сразу же усваивается благодаря уже имеющимся знаниям в ООП, и тому, что его возможности не конфликтуют, а дополняют опыт, накопленный при работе с другим языком программирования.

Если же программист только начинает свой путь в области разработки ПО, то Python станет идеальным «вводным» языком программирования. Благодаря своей лаконичности он позволит быстрее овладеть синтаксисом языка, а отсутствие «наследства» в виде формировавшихся на протяжении многих лет аксиом поможет быстро освоить ООП. В силу этих факторов «кривая обучения» Python будет довольно короткой, и программист сможет перейти от учебных примеров к коммерческим проектам [13].

С#



Рис.2. Логотип С#

На сегодняшний момент язык программирования С# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли (рис. 2). В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программ до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей.

По сравнению с другими языками С# достаточно молодой, но в то же время он уже прошел большой путь. Первая версия языка вышла вместе с релизом Microsoft Visual Studio .NET в феврале 2002 года. Текущей версией языка является версия С# 7.0, которая вышла в 7 марта 2017 года вместе с Visual Studio 2017.

С# является языком с Си-подобным синтаксисом и близок в этом отношении к С++ и Java. Поэтому, если вы знакомы с одним из этих языков, то овладеть С# будет легче.

C# является объектно-ориентированным и в этом плане много перенял у Java и C++. Например, C# поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. И C# продолжает активно развиваться, и с каждой новой версией появляется все больше интересных функциональностей, как, например, лямбды, динамическое связывание, асинхронные методы и т.д[12].

F#



Рис.3. Логотип F#

F# впервые увидел свет в 2005 году в MicrosoftResearch. Это мультипарадигмальный язык для платформы Microsoft .NET, нацеленный преимущественно на функциональное программирование, имеющий, тем не менее, возможность создания объектно-ориентированного и императивного кода (рис. 3).

Несмотря на своё название, F# не является развитием C#, напротив будучи скорее его полной противоположностью. Из общего у них разве что разработчик, да возможность реализации всех трёх парадигм программирования, однако приоритеты у них принципиально разные.

Как и другие функциональные языки, F# нацелен на упрощение работы с BigData и многопоточность, то есть востребован в науке, в создании искусственного интеллекта, там, где используются динамические базы данных, а

также для улучшенного использования ресурсов многоядерных процессоров. Иначе говоря, F# можно использовать практически везде.

Впрочем, F# - не сладкая вата, а живой язык со своими недостатками. Вот парочка из них:

1. Слабая поддержка мобильных платформ. Для разработчиков Android и iOS приложений в последнее время есть определенные позитивные изменения, но применять F# для этих целей по-прежнему неудобно;
2. Малочисленное сообщество разработчиков. Говоря языком цифр, их примерно в 100 раз меньше, чем программистов на C#.

Java



Рис. 4. Логотип Java

Язык Java зародился как часть проекта создания передового программного обеспечения (ПО) для различных бытовых приборов (рис. 4). Реализация проекта была начата на языке C++, но вскоре возник ряд проблем, наилучшим средством борьбы с которыми было изменение самого инструмента - языка программирования. Стало очевидным, что необходим платформо-независимый язык программирования, позволяющий создавать программы, которые не приходилось бы компилировать отдельно для каждой архитектуры и можно было бы использовать на различных процессорах под различными операционными системами. Язык Java потребовался для создания интерактивных продуктов для сети Internet. Фактически, большинство архитек-

турных решений, принятых при создании Java, было продиктовано желанием предоставить синтаксис, сходный с Си и Си++. В Java используются практически идентичные соглашения для объявления переменных, передачи параметров, операторов и для управления потоком выполнением кода. В Java добавлены все хорошие черты C++.

Три ключевых элемента объединились в технологии языка Java:

- Java предоставляет для широкого использования свои апплеты (applets) — небольшие, надежные, динамичные, не зависящие от платформы активные сетевые приложения, встраиваемые в страницы Web. Апплеты Java могут настраиваться и распространяться потребителям с такой же легкостью, как любые документы HTML.

- Java высвобождает мощь объектно-ориентированной разработки приложений, сочетая простой и знакомый синтаксис с надежной и удобной в работе средой разработки. Это позволяет широкому кругу программистов быстро создавать новые программы и новые апплеты.

- Java предоставляет программисту богатый набор классов объектов для ясного абстрагирования многих системных функций, используемых при работе с окнами, сетью и для ввода-вывода. Ключевая черта этих классов заключается в том, что они обеспечивают создание независимых от используемой платформы абстракций для широкого спектра системных интерфейсов [14].

Ruby

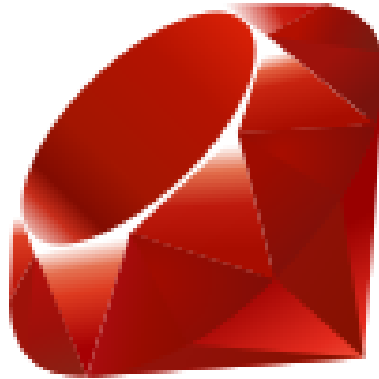


Рис.5. Логотип Ruby

Ruby - интерпретируемый язык высокого уровня для быстрого и удобного объектно-ориентированного программирования (рис. 5). Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, «сборщиком мусора» и многими другими возможностями. Ruby близок по особенностям синтаксиса к языкам Перл и Эйфель, по объектно-ориентированному подходу к Smalltalk. Также некоторые черты языка взяты из Python, Лисп, Dylan и CLU. Кроссплатформенная реализация интерпретатора языка является полностью свободной. Последней является версия 1.9.0, вышедшая 25 декабря 2007.

Возможности Ruby.

Имеет лаконичный и простой синтаксис, частично разработанный под влиянием Ада, Эйфель и Python.

Позволяет обрабатывать исключения в стиле Java и Python.

Позволяет переопределять операторы, которые на самом деле являются методами.

Полностью объектно-ориентированный язык программирования. Все данные в Ruby являются объектами в понимании Smalltalk. Единственное исключение — управляющие конструкции, которые в Ruby в отличие от Smalltalk не являются объектами. Например, число «1» — это экземпляр

класса `Fixnum`. Также поддерживается добавление методов в класс и даже в конкретный экземпляр во время выполнения программы.

Не поддерживает множественное наследование, но вместо него может использоваться концепция «примесей», основанная в данном языке на механизме модулей.

Содержит автоматический сборщик мусора. Он работает для всех объектов Ruby, в том числе для внешних библиотек.

Создавать расширения для Ruby на Си очень просто частично из-за сборщика мусора, частично из-за несложного и удобного API.

Поддерживает замыкания с полной привязкой к переменным. Поддерживает блоки кода (код заключается в `{ ... }` или `do ... end`). Блоки могут использоваться в методах или преобразовываться в замыкания.

Целые переменные в Ruby автоматически конвертируются между типами `Fixnum` (32-разрядные) и `Bignum` (больше 32 разрядов) в зависимости от их значения, что позволяет производить целочисленные математические расчёты с бесконечной точностью.

Не требует предварительного объявления переменных, но для интерпретатора желательно, чтобы переменным присваивалось пустое значение `nil` (тогда интерпретатор знает, что идентификатор обозначает переменную, а не имя метода). Язык использует простые соглашения для обозначения области видимости. Пример: просто `var` — локальная переменная, `@var` — переменная экземпляра (член или поле объекта класса), `@@var` — переменная класса, `$var` — глобальная переменная.

В Ruby непосредственно в языке реализованы многие шаблоны проектирования, так, например, «одиночка» (`singleton`) может быть (хотя и не обязан) реализован добавлением необходимых методов к одному конкретному объекту.

Может динамически загружать расширения, если это позволяет операционная система.

Имеет независимую от ОС поддержку невытесняющей многопоточности.

Перенесён на множество платформ. Он разрабатывался на GNU/Linux, но работает на многих версиях Unix, DOS, Microsoft Windows (в частности, Win32), Mac OS, BeOS, OS/2 и т. д [13].

Так, высокоуровневые языки стремятся не только облегчить решение сложных программных задач, но и упростить портирование программного обеспечения. Использование разнообразных трансляторов и интерпретаторов обеспечивает связь программ, написанных при помощи языков высокого уровня, с различными операционными системами программируемыми устройствами и оборудованием, и, в идеале, не требует модификации исходного кода (текста, написанного на высокоуровневом языке) для любой платформы.

Такого рода оторванность высокоуровневых языков от аппаратной реализации компьютера помимо множества плюсов имеет и минусы. В частности, она не позволяет создавать простые и точные инструкции к используемому оборудованию. Программы, написанные на языках высокого уровня, проще для понимания программистом, но менее эффективны, чем их аналоги, создаваемые при помощи низкоуровневых языков. Одним из следствий этого стало добавление поддержки того или иного языка низкого уровня (язык ассемблера) в ряд современных профессиональных высокоуровневых языков программирования.

1.2. Обзор образовательных курсов по программированию на языке Python

Python - интерпретируемый, объектно-ориентированный высокоуровневый язык программирования с динамической семантикой. Встроенные высокоуровневые структуры данных в сочетании с динамической типизацией и связыванием делают язык привлекательным для быстрой разработки приложений (RAD, RapidApplicationDevelopment). Кроме того, его можно использовать в качестве сценарного языка для связи программных компонентов. Синтаксис Python прост в изучении, в нем придается особое значение читаемости кода, а это сокращает затраты на сопровождение программных продуктов. Python поддерживает модули и пакеты, поощряя модульность и повторное использование кода. Интерпретатор Python и большая стандартная библиотека доступны бесплатно в виде исходных и исполняемых кодов для всех основных платформ и могут свободно распространяться.

1. Курс программирования на языке Python в виде тетрадок IPython

Курс состоит из 5 уроков: обзор средств разработки, введение в язык Python, 2 урока про структуры данных (Python и не только) и некоторые алгоритмы и один урок про функции и рекурсию. Да, не затронута тема ООП и кучи всего полезного, но работа над курсом идет, репозиторий будет обновляться.

Цели данного курса:

- Познакомить с основами языка Python
- Познакомить с основными структурами данных
- Дать навыки, необходимые для разработки простых алгоритмов

Тетрадки IPython выбраны в качестве основного средства подачи материала из-за того, что в них можно совмещать текст, картинки, формулы и код. В ка-

честве демонстрации в репозитории приведена тетрадка про деревья решений и их реализацию в библиотеке машинного обучения Scikit-learn. В материалах первого урока описано, как собственно этими тетрадками пользоваться[8].

2. Майк МакГрат "Программирование на Python для начинающих"

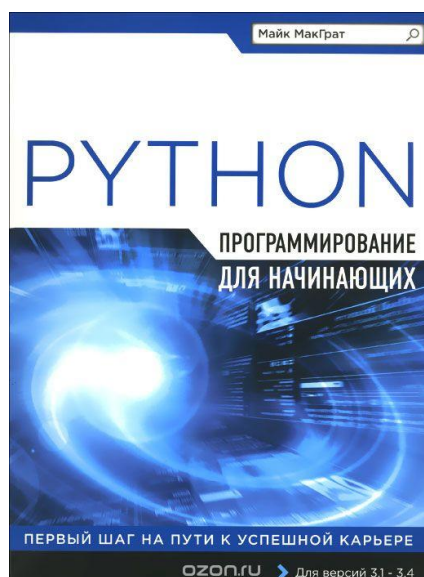


Рис. 6. Программирование на Python для начинающих

В руководстве подробно рассмотрены все основные особенности языка: описание технологии, структура, синтаксис (рис. 6). Так, как книга предназначена для начинающих и является практическим базовым самоучителем, то в ней приводятся все этапы начиная от установки интерпретатора Python на разные платформы операционных систем, создания первой программы 'HelloWorld!' до примеров разработки многофункциональных WEB-приложений.

Примеры кода, которые вы найдете в книге, рассчитаны на начинающих программистов они показывают пошагово, как создавать приложения на языке Python. Результат работы программного кода - демонстрируется на скриншотах. Руководство "Программирование на Python для начинающих" идеально подойдет для начального обучения, в нем собраны только самые основные данные по языку [10].

3. Python



Рис. 7. Курс «Python»

На курсе Вы освоите синтаксис языка Python, получите базовые навыки структурного и процедурного программирования, познакомитесь с базовыми структурами данных языка Python (рис. 7). Вы научитесь устанавливать и настраивать среду разработки, применять базовые конструкции Python, создавать модули и пакеты, пользоваться основными структурами данных, выполнять основные операции ввода/вывода. Также Вы получите необходимую подготовку для изучения объектно-ориентированного программирования на языке Python.

Программа курса соответствует требованиям профессионального стандарта «Программист».

Курс ориентирован на Python версии от 3.3 и более новые. В необходимых случаях отмечаются особенности версии 2.7 [7].

4. Программирование и научные вычисления на языке Python

Цель этого курса — научиться писать программы, решающие научные задачи. Язык программирования Python обладает ясным и понятным синтаксисом, и потому легко учится и хорошо подходит для введения в программирование. Это позволит нам не отвлекаться от решаемой задачи на особенности языка и их объяснение, позволит естественным образом последовательно вводить новые инструменты. Язык близок к MATLAB и хорош для программирования математических вычислений. К тому же Python умеет работать с такими языками как Fortran, C и C++, которые уже широко используются в

научных расчетах. Кроме всего прочего, большинство продуктов и модулей, написанных на Python, распространяются бесплатно, в том числе и все те инструменты, которыми мы будем пользоваться в ходе наших уроков.

Примеры данного курса интегрируют программирование с прикладным вычислениями в математике, физике, биологии и экономике. Предполагается, что читатель знаком с математическим анализом функции одной переменной и базовыми законами физики, тогда примеры окажутся более наглядными. В свою очередь, знание языка Python или умение программировать на любом другом языке совсем не предполагается и язык изучается одновременно с решением задач. Знать же английский язык, хотя бы на базовом уровне, будет очень неплохо. Вы и сами это знаете.

Многие книги по программированию фокусируют свое внимание исключительно на особенностях языка программирования. Однако, чтобы писать программы, надо думать как программист. Это означает, что надо находить решение на поставленную задачу. В этом курсе в каждом новом примере определяется новая задача и находится ее решение в виде программного кода [1].

5. *Python. Продвинутый курс*

Рассматриваются более глубокие аспекты языка программирования Python. Python является чрезвычайно мощным динамический языком программирования, который используется в самых разнообразных прикладных областях, включая телекоммуникации. Python часто сравнивают с PHP, Perl, Ruby, Scheme или Java.

Цели

- Изучить продвинутые функции Python.
- Объяснить, почему Python® настолько популярен и почему стоит его изучать?
- Рассмотреть обширную пользовательскую базу.

- Изучить краткое повторение типов данных, написания функций, управления потоками, классов, постепенное добавление более продвинутого использования.
- Научиться применять Python® на практике.

Учебные темы

- Почему Python так популярен и где он используется?
- Список программ, о которых вы не знали, что они написаны на языке Python.
- Первый запуск Python.
- Большой список интерпретаторов и дистрибутивов.
- Повторение + расширенное применение типов данных: логические, числовые, строки, форматирование строк, списки, кортежи, словари, диапазоны, итераторы, объект, файлы, файлоподобные объекты, исключения, указатели/ссылки, внутренний размер объекта.
- Ключевые слова. Что это, что с помощью них можно сделать.
- Функции и лямбды.
- Краткое повторение: управление потоками - if, elif, else, while, for.
- Классы и объекты. "Магические" методы.
- Библиотеки. Популярные стандартные и нестандартные библиотеки. Как установить нестандартную библиотеку. Virtualenv.
- Стандартные модули: наиболее часто используемые модули и функции, с примерами: builtin, time, timeit, random, regex, pprint, traceback, unittest, персистентность и сериализация, configparser.
- Рекомендуемый стиль написания кода – PEP-8.
- Дружественные IDEs. Хостинг кода Python.
- Практическое применение Python. Мощные однострочные скрипты. Нестандартные библиотеки с примерами. Доступ к FTP, Telnet, SSH, веб-серверы, HTTPS, потоки, веб-автоматизация, веб-сборщики.

- Как написать библиотеку. Где ее разместить и как использовать [3].

6. Занятия по программированию на *Python* в ТУСУР

Курс не требует предварительного знакомства с программированием, обучение проходит с нуля. Во время обучения участники еженедельно будут писать небольшие игры с простой графикой, используя только что полученные знания. На последней неделе обучающиеся напишут аркадную игру пинг-понг для двоих игроков.

Полученные навыки будут применимы не только в области написания игр, но также и в научных вычислениях, веб-программировании, системном администрировании и многих других областях.

Курс основан на материале от образовательной площадки Coursera, необходимо начальное знание английского языка. В ходе курса будет оказываться поддержка в улучшении навыков технического английского [4].

ГЛАВА 2. РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА «ЯЗЫК PYTHON»

ВЫПИСКА

из протокола № 11 заседания кафедры мультимедийной дидактики и информационных технологий обучения Пермского государственного педагогического университета
от 14 июня 2017 ГЮ

ПРИСУТСТВОВАЛИ: зав. кафедрой., д.п.н., профессор Е.В. Оспенникова; канд. тех. наук, профессор О.И. Мухин, кандидат физ.-мат. наук, доцент Е.А. Еремин; кандидат физ.-мат. наук, кандидат физ.-мат. наук, доцент Д.В. Баяндин; доцент, канд. пед. наук И.В. Ильин; доцент, канд. тех. наук О.Ю. Вологжанин; доцент, канд. пед. наук А.А. Оспенников, АСС. Д.А. Голубев, асп. В.В. Аспидов; асп. В.В. Васенев; асп. Д.А. Терехин

СЛУШАЛИ: Руководителей выпускных квалификационных работ студентов, обучающихся по направлению 09.02.03 «Информационные технологии в образовании», о наличии в тексте ВКР сведений, имеющих действительную и/или потенциальную коммерческую ценность.

В соответствии с п.38 Приказа Министерства образования и науки РФ от 29.06.2015 № 636 «Об утверждении Порядка проведения государственной итоговой аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры» доступ лиц к текстам выпускных квалификационных работ должен быть обеспечен в соответствии с законодательством Российской Федерации, с учетом изъятия производственных, технических, экономических, организационных и других сведений, в том числе о результатах интеллектуальной деятельности в научно-технической сфере, о способах осуществления профессиональной деятельности, которые имеют действительную или потенциальную коммерческую ценность в силу неизвестности их третьим лицам, в соответствии с решением правообладателя. 9

ПОСТАНОВИЛИ: изъять из текста выпускной квалификационной работы студента Заболотских Евгения Дмитриевича страницы с 21 по 38 при размещении работы в электронно-библиотечной системе ПГГПУ <http://vkr.pspu.ru/>.

Зав. кафедрой, МД и ИТО, профессор

Оспенникова Е.В.

14 июня 2017 года

Заключение

По итогам выполненной работы получены следующие результаты:

1. Рассмотрены современные языки программирования высокого уровня. Выделены особенности и свойства этих языков.
2. Проанализированы образовательные курсы по программированию на языке Python. Составлена характеристика и описание учебных курсов.
3. Выделена структура учебного модуля «Язык Python». Описаны элементы интерфейса и их назначение.
4. Разработан учебный модуль «Язык Python» в программной среде IDLE с использованием языка Python.

Модуль «Язык Python» предназначен для изучения интерпретируемого, высокоуровневого языка Python и закрепления полученного материала.

Библиографический список

1. An Introduction to Interactive Programming in Python by Rice University / on Coursera [Электронный ресурс]. - URL: <https://www.coursera.org/course/interactivepython> (Дата обращения: 05.05.2017)
2. Programming for Everybody by Charles Severance, University of Michigan / on Coursera [Электронный ресурс]. - URL: <https://class.coursera.org/pythonlearn-001> (Дата обращения: 05.08.2017)
3. TheErikPython IDE [Электронный ресурс]. - URL: <http://eric-ide.python-projects.org> (Дата обращения: 16.05.2017)
4. Tutorialspoint / Python / XMLProcessing [Электронный ресурс]. - URL: http://www.tutorialspoint.com/python/python_xml_processing.htm (Дата обращения: 04.29.2017)
5. Ашарина И.В. Объектно-ориентированное программирование в C++ [Электронный ресурс]: учебное пособие/ Ашарина И.В.— Электрон. текстовые данные.— М.: Горячая линия - Телеком, 2012.— 320 с.— Режим доступа: <http://www.iprbookshop.ru/12008>.— ЭБС «IPRbooks»
6. Баженова И. Ю. Язык программирования Java. - М.: Диалог-МИФИ, 2008. <http://www.biblioclub.ru/book/54745/>
7. Вирт Н. Алгоритмы и структуры данных М.: ДМК Пресс, 2010. - 272 с. http://www.biblioclub.ru/86483_Algoritmy_i_struktury_dannykh.html
8. Дэвид Бизли. Python. Подробный справочник [Электронный ресурс]: учебное пособие. <http://progbook.net/python/6601-python-podrobnyj-spravochnik.html>
9. Кознов Д.В. Введение в программную инженерию [Электронный ресурс]/ Кознов Д.В.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2009.— 189 с.— Режим доступа: <http://www.iprbookshop.ru/16697>.— ЭБС «IPRbooks»
10. Корнев Д. В. Unpus - декомпилятор языка python // “СПИСОК- 2009”, Материалы межвузовской научной конференции по проблемам информатики. — Екатеринбург: УрО РАН, 2009. — С. 30–33.
11. ЛутцМ.ИзучаемPython, 4-е издание- М.: Диалог-МИФИ, 2010.<http://www.symbol.ru/alphabet/816603.html>
12. Мишенин А. И. Сборник задач по программированию. Учебное пособие - М.: Финансы и статистика, 2009. <http://www.biblioclub.ru/book/86065/>

13. Смирнов А. А. Технологии программирования. Учебно-практическое пособие - М.: Евразийский открытый институт, 2011.
<http://www.biblioclub.ru/book/90777/>
14. Терехов А.Н. Технология программирования [Электронный ресурс]: учебное пособие/ Терехов А.Н.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий (ИНТУИТ), 2007.— 148 с.— Режим доступа:
<http://www.iprbookshop.ru/22447>.— ЭБС «IPRbooks»
15. Хорев П.Б. Объектно-ориентированное программирование: учеб. пособие для студ. Учреждений высшего профессионального образования / П.Б. Хорев. – 3-е изд., испр. М.: Издательский центр «Академия», 2011. – 448с.