

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕПАРТАМЕНТ ОСВІТИ І НАУКИ
ЗАПОРІЗЬКОЇ ОБЛАСНОЇ ДЕРЖАВНОЇ АДМІНІСТРАЦІЇ

КОМУНАЛЬНИЙ ЗАКЛАД «ЗАПОРІЗЬКИЙ ОБЛАСНИЙ ЦЕНТР
НАУКОВО-ТЕХНІЧНОЇ ТВОРЧОСТІ УЧНІВСЬКОЇ МОЛОДІ «ГРАНІ»
ЗАПОРІЗЬКОЇ ОБЛАСНОЇ РАДИ

СХВАЛЕНО

Протокол засідання науково
методичної ради КЗ «ЗОШПО»
Запорізької обласної ради

« 14 » 06. 2020 № 5
Голова НМР КЗ «ЗОШПО» ЗОР
О.В.Варецька



ЗАТВЕРДЖЕНО

Наказ Департаменту освіти і науки
Запорізької обласної державної
адміністрації

« 13 » 08. 2020 № 320
Тимчасово виконуючий обов'язки
директора Департаменту

В.М.ЗАХАРЧУК



**Навчальна програма з позашкільної освіти
науково-технічного напрямку**

«Програмування мовою Python»

початковий рівень, перший та другий роки навчання

м. Запоріжжя – 2020

Укладач:

Луцишин Максим Віталійович – керівник гуртка «Програмування мовою Python» відділу STEM-освіти комунального закладу «Запорізький обласний центр науково-технічної творчості учнівської молоді «Грані» Запорізької обласної ради.

Рецензенти:

Чаусовський Григорій Олександрович – кандидат технічних наук, доцент кафедри загальної та прикладної екології і зоології Запорізького національного університету, керівник гуртка «Біофізика»

Смирнова Анна Василівна – завідувач відділу STEM - освіти, керівник гуртка «Основи робототехніки та комп'ютерного моделювання»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Розвиток сучасних технологій неможливий без використання комп'ютерної техніки та програмного забезпечення. Підготовка фахівців у галузі інформаційних технологій вимагає великих знань та навичок володіння комп'ютерною технікою, а також знання основ алгоритмізації та програмування мовами високого рівня.

Python – це універсальна мова, що широко використовується в усьому світі для самих різних цілей – бази даних і оброблення текстів, вбудовування інтерпретатора в ігри, програмування GUI, швидке створення прототипів (RAD) програмування Internet і Web додатків – серверних (CGI), клієнтських (роботи), Webсерверів і серверів додатків.

Зручність мови Python оснований на тому, що вона є мовою високого рівня, має набір конструкцій структурного програмування та підтримує модульність. Гнучкість та універсальність мови Python забезпечує її широке розповсюдження.

Уже у молодшому шкільному віці діти захоплюються програмуванням (особливо графічних та ігрових програм). Учні такої вікової категорії потребують цікавого та веселого навчання, тому вивчення програмування відбувається за допомогою програмних засобів малювання.

Навчальна програма реалізується у гуртках науково-технічного напрямку інформаційно-технічного профілю та спрямована на вихованців 10-12 років. На опрацювання навчального матеріалу відводиться 144 год. на рік (по 4 год. на тиждень).

Програму гуртка «Програмування мовою Python» розроблено на 2 роки навчання початкового рівня :1-й рік навчання (вихованців віком 10-11 років), 2-й рік навчання (вихованців віком 11-12 років).

Кількісний склад навчальної групи – 15 учнів. Навчальна програма пройшла адаптацію на заняттях гуртку «Програмування мовою «Python» у КЗ «ЗОЦ НТТУМ «Грані» ЗОР.

Метою навчальної програми є формування базових понять структурного програмування та розвиток інтелектуальних, емоційно-ціннісних, креативних та логістичних здібностей у вихованців гуртка.

Основні завдання полягають у формуванні таких **компетентностей**:

- *пізнавальної*, зміст якої полягає у розвитку базових знань з основ математики, програмування, логіки, навчання основним прийомам та способам розробки програмних додатків, базових алгоритмів, вмінні застосовувати програмний продукт для полегшення повсякденних задач.

- *практичної*, зміст якої полягає у створенні загальних вмінь: виконувати завдання відповідно до поставленої мети, планувати діяльність, аналізувати та оцінювати її результат.

- *творчої*, яка спрямована на створення умов для активного розвитку всіх психічних процесів, зокрема логічного мислення; репродуктивної та творчої уяви; образної, рухової та мовно-логічної пам'яті; розвитку навичок користування комп'ютером та розподілу поставленої задачі на групу простих завдань.

- *соціальної*, яка передбачає формування морально-етичних цінностей та виховання таких базових якостей особистості, як самостійність, цілеспрямованість, наполегливість, креативність; доводити розпочату роботу до кінця, працювати поряд та разом з дорослими і однолітками.

Навчальною програмою передбачається використання мови програмування високого рівня Python, яка є однією з найпотужніших сучасних мов програмування загального призначення. Python – нова мова програмування, потенціал якої лише розкривається і ще не використовується у повній мірі. До того ж це проста у вивченні мова програмування, а готовий код легко читати, що дозволяє краще засвоїти навички з побудови алгоритмів.

Середовище розробки дозволяє одразу отримувати результат роботи, аналізувати та легко виправляти помилки у разі їх виникнення. У мові Python присутній спеціальний графічний модуль turtle, що дозволяє вивчати програмування при створенні простих малюнків. Це викликає в учнів підвищену цікавість, на відміну від складних математичних операцій.

Програма «Програмування мовою «Python» сприяє практичному опануванню конструкцій мови програмування Python, правил і методів складання, редагування та використання програм у середовищі програмування, опанування навичок програмування, проектування та створення програм, розвиток інтелектуальних та творчих здібностей вихованців для подальшого вивчення поглибленого курсу інформаційних технологій та програмування для отримання відповідної ІТ-спеціальності.

Python підтримує структурне, узагальнене, об'єктно-орієнтоване, функціональне і аспектно-орієнтоване програмування. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

Python підтримує більшість активно використовуваних платформ. Він поширюється під вільною ліцензією, що дозволяє безкоштовно використовувати його без обмежень в будь-яких додатках. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін. Ще

однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо.

Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

Навчальну програму побудовано за лінійним принципом «від простого до складного». Послідовність викладення матеріалу є аналогічною послідовності викладення матеріалу в книжці Джейсона Бріггса «Python для дітей», яка написана автором спеціально для дітей молодшого шкільного віку.

Завдяки такій послідовності тем учні із задоволенням сприймають новий матеріал, легко його засвоюють та мають змогу проявляти ініціативу та вирішувати певну задачу багатьма різними способами, використовуючи раніше засвоєний матеріал, що також сприяє результативному закріпленню та систематизації знань в цілому.

Формою контролю за результативністю навчання є практичні, підсумкові роботи, участь у конкурсах, творчі звіти.

Навчання в гуртку не передбачає спеціальної підготовки та знань. Навчальний матеріал програми викладено за принципом доступності та відповідності його обсягу віковим особливостям вихованців.

Програма є орієнтовною. За необхідності керівник гуртка може внести до програми зміни, які не повинні впливати на загальний зміст навчальної програми та кількість навчальних годин. Незмінними мають залишатися мета, завдання і прогнозований результат освітньої діяльності.

**Початковий рівень, перший рік навчання
НАВЧАЛЬНО-ТЕМАТИЧНИЙ ПЛАН**

№ п/п	Тема	Кількість годин		
		теоретичні	практичні	усього
1.	Вступ	2	–	2
2.	Загальний огляд мов програмування. Ознайомлення із середовищем мови Python	4	4	8
3.	Прості розрахунки та змінні	2	6	8
4.	Типи даних мови Python	6	12	18
5.	Використання засобів модуля черепахи	8	28	36
6.	Запис логічних умов та конструкції if	6	12	18
7.	Циклічні конструкції	10	16	26
8.	Повторне використання коду	10	16	26
9.	Підсумок	2	–	2
	Разом	50	94	144

ЗМІСТ ПРОГРАМИ

1. Вступ (2 год.)

Теоретична частина: Мета, завдання та зміст роботи гуртка. Організаційні питання. Правила безпеки життєдіяльності. Ознайомити учнів з комп'ютерним класом та робочим місцем. Навчити вмикати комп'ютер, завантажувати операційну систему.

2. Загальний огляд мов програмування. Ознайомлення із середовищем мови Python (8 год.)

Теоретична частина: Загальний огляд мов програмування, їх класифікація та можливості. Мови програмування, транслятори, компілятори, інтерпретатори, різниця між ними. Мова програмування *Python*. Загальна характеристика, історія виникнення та можливості.

Практична частина: Встановлення останньої версії *Python*.

3. Прості розрахунки та змінні (8 год.)

Теоретична частина: Консоль *Python*. Shell (IDLE). Різниця між консоллю *Python* і Shell (IDLE).

Практична частина: Завантаження останньої версії *Python* та її встановлення. Відкриття програми-прикладу мовою *Python* і запускання її з Shell. Створення простих програм. Запускання їх з командного рядка та з Shell (відкривати файл і запускати його в середовищі Shell). У Shell здійснювати обчислення з використанням різних операторів: + - * /. Аналіз використання дужок та їх вплив на результати розрахунків.

4. Типи даних мови Python(18 год.)

Теоретична частина: Використання констант та змінних для зберігання даних. Різниця між місцем у пам'яті для зберігання фактичного значення та назвою змінної, яка є міткою, що вказує на значення. Цілі та дійсні змінні. Різниця між числом і рядком. Використання списків та кортежів. Різниця між кортежем і списком. Логічні змінні.

Практична частина: Розробка програм з використанням констант і змінних. Розробка лінійних програм із використанням числових, логічних, символічних змінних, кортежів та списків.

5. Використання засобів модуля черепахи (36 год.)

Теоретична частина: Базове поняття модуля (як одиниці коду). Імпорт модулів. Модуль черепахи (*turtle*). Поняття пікселя, лінії, основних графічних примітивів. Основні кути для повороту черепахи (45, 90, 135 тощо). Основні команди черепахи.

Практична частина: Відкриття Shell й імпортування модуля черепахи (*turtle*). Малювання черепахою ліній. Повороти черепахи на різні кути. Малювання квадратів різних розмірів. Малювання квадратів під кутом. Використання команд «вгору» (*up ()*) та «вниз» (*down ()*). Малювання кіл (*circle ()*). Зміна кольору пера (*pencolor ()*) та товщини пера (*pensize ()*). Малювання кругів (*dot ()*) різного кольору. Зміна форми черепахи (*shape ()*) та швидкості малювання (*speed ()*). Використання практичних робіт на побудову зображень за зразком.

6. Запис логічних умов та конструкції if (8 год.)

Теоретична частина: Умови та їх запис. Реалізація питання в програмі за допомогою *if*-конструкції. Структура *if*-конструкції. Поняття блоків коду, що входять до складу *Python*-конструкцій (записуються з відступом). Команди *if* та *elif*. Об'єднання умов. Змінна без значення – *None*. Базове розуміння користувацького вводу в *Python*. Функції *int()* та *str()*, які перетворюють рядки на числа і числа на рядки.

Практична частина: Розробка та використання програм з *if*-конструкціями та конструкціями *elif*. Використання в програмах користувачького вводу та функцій *int()* та *str()*. Використання *if*-конструкцій під час побудови зображень за зразком.

7. Циклічні конструкції (26 год.)

Теоретична частина: Використання циклів для повторюваних завдань. Групування команд, які потрібно дублювати, у блоки коду і розміщення їх у циклах. Види циклів: *for* і *while*. Використання циклу *for* та функції *range()*. Використання списків у циклах. Використання вкладених циклів. Використання циклу *while* та команди *break* для примусового завершення циклу.

Практична частина: Розробка та реалізація найпростіших програм з циклами. Розробка та реалізація графічних програм з циклами.

8. Повторне використання коду (26 год.)

Теоретична частина: Застосування функцій користувача. Структура та побудова функцій користувача. Параметри, змінні та область видимості. Повернення результатів функціями (*return ()*). Застосування модулів для угруповання функцій. Огляд основних модулів, які входять у комплект *Python*. Використання функцій й модуля *sys* та модуля *time*.

Практична частина: Розробка та виконання програм з використанням функцій. Розробка графічних програм з використанням функцій.

9. Підсумок (2 год.)

Теоретична частина: Підведення підсумків роботи гуртка. Аналіз вивченого матеріалу.

ПРОГНОЗОВАНИЙ РЕЗУЛЬТАТ

Вихованці повинні розуміти:

- різницю між програмою та мовою програмування;
- різницю між консоллю *Python* і *Shell (IDLE)*;
- різницю між місцем у пам'яті, яке використовується для зберігання фактичного значення та назвою змінної, яка є міткою, що вказує на значення;
- різницю між числом і рядком;
- різницю між кортежем і списком;
- різницю між трансляторами, компіляторами, інтерпретаторами;
- поняття модуля (як одиниці коду);
- клас *Pen* модуля *turtle* та його об'єкти.

Вихованці повинні знати:

- основні типи змінних;

- що таке лінійні програми;
- що таке блоки коду, що входять до складу *Python*-конструкцій;
- структуру *if*- та *elif*-конструкцій;
- що таке користувацький ввід у мові *Python*;
- що таке функції користувача та коли потрібно їх застосовувати;
- що таке параметри, змінні та область видимості.

Вихованці повинні вміти:

- створювати прості програми, запускати їх на виконання з *Shell*;
- використовувати *Shell* (або консоль) як простий калькулятор;
- використовувати списки та кортежі;
- імпортувати модуль черепахи (*turtle*) і малювати фігури та зображення;
- створювати умови з допомогою *if*-конструкцій;
- створювати умовні конструкції та поєднувати їх;
- групувати команди, які потрібно повторювати, у блоки коду і розміщувати їх у циклах.

Вихованці мають набути досвід:

- щодо уявлень про будову програмних продуктів, розуміння принципу роботи, етапів та технології створення; засвоєння потрібних слів та правильне їх вживання;
- щодо вміння виконувати завдання відповідно до поставленої мети, планування своєї діяльності; вміння аналізувати та оцінювати результати своєї діяльності;
- щодо формування конструкторського та логічного мислення; репродуктивної та творчої уяви; образної та словесно-логічної пам'яті,
- щодо розвитку морально-етичних цінностей дитини; виховання таких якостей особистості як: самостійність, цілеспрямованість, наполегливість, креативність; доведення розпочатої роботи до кінця; навички працювати разом з дорослими та однолітками.

**Початковий рівень, другий рік навчання
НАВЧАЛЬНО-ТЕМАТИЧНИЙ ПЛАН**

№ п/п	Тема	Кількість годин		
		теоретичні	практичні	усього
1.	Вступ	2	–	2
2.	Відновлення та вдосконалення навичок використання основних конструкцій мови	6	18	24
3.	Поняття класів та об'єктів	18	20	38
4.	Об'єкти та класи в модулі turtle	12	16	28
5.	Вмонтовані функції Python	4	14	18
6.	Робота з файлами	2	16	18
7.	Створення зображень та анімації засобами модуля tkinter	12	48	60
8.	Створення гри «Стрибаючий м'яч»	10	16	26
9.	Підсумок	2	–	2
	Разом	68	148	216

ЗМІСТ ПРОГРАМИ

1. Вступ (2 год.)

Теоретична частина: Мета, завдання та зміст роботи гуртка. Організаційні питання. Повторення правил безпеки життєдіяльності. Загальний огляд модулів мови *Python* та їх можливостей.

2. Відновлення та вдосконалення навичок використання основних конструкцій мови (24 год.)

Теоретична частина: Типи даних мови *Python*. Опис логічних умов та конструкцій *if* та *elif*. Циклічні конструкції *while* та *for*. Повторне використання коду. Використання засобів модуля черепахи.

Практична частина: Виконання завдань на використання різних типів даних мови *Python*, логічних умов та конструкції *if*, циклічних конструкцій та функцій користувача. Побудова графічних зображень за допомогою засобів модуля *turtle*.

3. Поняття класів та об'єктів (38 год.)

Теоретична частина: Розподіл предметів на класи та підкласи. Поняття об'єкта. Нащадки та предки. Використання функцій для створення характеристики класу. Створення об'єктів класів. Функції, що викликають інші функції. Аргумент *self*. Успадкування функцій. Ініціалізація об'єктів.

Практична частина: Створення класу *Things* (предмети), розділення його на підкласи (живі, неживі). Створення в підкласі «живі» підкласів «тварини», «савці», «жирафи», у підкласі «неживі» – підклас «тротуари». Створення кількох об'єктів класів. Побудова функцій, що викликають функції. Створення функції ініціалізації для класу жирафів.

4. Об'єкти та класи в модулі *turtle* (28 год.)

Теоретична частина: Клас *Pen* модуля *turtle* та його об'єкти-черепашки. Функції класу *Pen* для реалізації команд черепашкам. Виклик функцій кожного об'єкта окремо, щоб черепашки пересувалися і малювали незалежно одна від одної.

Практична частина: Створення кількох різних об'єктів-черепашок та виконання різних команд.

5. Вмонтовані функції *Python* (18 год.)

Теоретична частина: Вмонтовані функції мови *Python*: *abs ()*, *float ()*, *sum ()*, *min ()*, *max ()*, *len ()*, *exec ()*, *dir ()*, *help ()*, *eval ()*, *range ()* та інші. Функції для роботи з файлами.

Практична частина: Розв'язування задач із вмонтованими функціями *Python*. Побудова зображень з використанням вмонтованих функцій.

6. Робота з файлами (18 год.)

Теоретична частина: Створення текстового файлу. Функція *open ()* для відкриття файлу на читання та запис. Атрибути *'r'* та *'w'*. Функції *read ()* та *write ()* для читання та запису даних у файл. Функція *close ()* для закривання файлу.

Практична частина: Створення текстових файлів та використання в програмах інформації з них.

7. Створення зображень та анімації засобами модуля tkinter (60 год.)

Теоретична частина: Опис можливостей модуля *tkinter*. Поняття іменованих аргументів. Створення кнопок (*Button*) та полотна (*Canvas*) для малювання. Малювання ліній (*create_line*) та прямокутників (*create_rectangle*). Робота з кольором. Використання *colorchooser.askcolor* (). Малювання дуг (*create_arc*). Малювання багатокутників (*create_polygon*). Відображення тексту на полотні (*create_text*). Виведення зображень на полотно (*create_image*). Використання функцій *move()*, *update()* та *sleep()* для створення простої анімації. Реакція об'єктів на події. Поняття ідентифікаторів та їх використання.

Практична частина: Побудова графічних зображень з використанням функцій модуля *tkinter*. Створення анімацій з використанням функцій модуля *tkinter*.

8. Створення гри «Стрибаючий м'яч» (26 год.)

Теоретична частина: Особливості побудови ігрових програм.

Практична частина: Створення гри «Стрибаючий м'яч», у якій потрібно створити класи рухів, переміщень, зіткнення з відскоком м'яча та анімацію ракетки.

9. Підсумок (2 год.)

Теоретична частина: Аналіз вивченого матеріалу.

ПРОГНОЗОВАНИЙ РЕЗУЛЬТАТ

Вихованці повинні розуміти:

- різницю в демонструванні та використанні чисел та рядків;
- чим відрізняються кортежі від списків;
- що таке словники;
- що таке модуль (як одиниця коду);
- як використовувати модулі та їх об'єкти;
- що таке класи та їх об'єкти.

Вихованці повинні знати:

- типи даних мови *Python*;
- основні конструкції мови *Python*;
- як застосовуються модулі для групування функцій;
- як використовуються класи для класифікації предметів;
- як у програмах створювати і використовувати об'єкти (екземпляри) класів;
- як клас-нащадок успадковує функції класів-предків;
- як викликати для об'єкта функції, визначені в його класі та в класах-предках і зберігати значення у властивостях об'єктів;
- як за допомогою аргумента *self* отримувати доступ до функцій і властивостей класу з інших його функцій.

Вихованці повинні вміти:

- використовувати вкладені цикли;
- будувати функції користувача;
- користуватися функціями стандартних модулів (*sys, time, random, math*);
- використовувати функції класу *Pen* для реалізації команд черепашкам;
- використовувати вбудовані функції мови *Python*;
- працювати із файлами;
- малювати на полотні геометричні фігури та зображення;
- створювати анімації;
- створювати прив'язку клавіш до викликів функцій відповідних класів.

Вихованці мають набути досвід:

- щодо уявлень про будову програмних продуктів, розуміння алгоритму роботи певної програми при її використанні, формування ідей щодо покращення програмного продукту, та шляхи їх виконання;
- щодо вміння виконувати завдання відповідно до поставленої мети, планування своєї діяльності; вміння аналізувати та оцінювати результати своєї діяльності;
- щодо конструкторського та логічного мислення; репродуктивної та творчої уяви; образної та словесно-логічної пам'яті;
- щодо морально-етичних цінностей дитини; виховання таких якостей особистості як: самостійність, цілеспрямованість, наполегливість, креативність; доведення розпочатої роботи до кінця; навички працювати разом з дорослими та однолітками.

ЛІТЕРАТУРА

1. Пістунов І.М. Електронна економіка. Том 1. Криптовалюта. Big Data [Електронний ресурс]: Навч. посібник/ І.М. Пістунов, О.П. Антонюк – / М-во освіти і науки України; Нац. Гірн. ун-т. – Д.: НГУ, 2017. – 133 с. Режим доступу: http://pistunovi.inf.ua/EE_KC_BD.pdf (дата звернення: 17.12.2017). – Назва з екрана. Виписка з протоколу №20 Вченої Ради ДВНЗ "НГУ"
2. Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова : Видавничо-поліграфічний центр "Київський університет" – К, 2014. – 640 с.
3. Пістунов І.М. Нейромережеві технології економіці та фінансах з розрахунками на комп'ютері [Електронний ресурс]: навч. посібн. / І.М. Пістунов, О.П. Антонюк ; Нац. гірн. ун-т. – Електрон. текст. дані. – Д. : НГУ, 2014. – 125 с. Сертифікат ДВНЗ "НГУ".
4. Марк Лутц. Программирование на Python / Пер. с англ. – 4-е изд. – СПб.: Символ-Плюс, 2011. – Т. I. – 992 с.
5. Марк Лутц. Программирование на Python / Пер. с англ. – 4-е изд. – СПб.: Символ-Плюс, 2011. – Т. II.
6. Дэвид М. Бизли. Python. Подробный справочник, 4-е издание. – Перевод с английского. – СПб.: Символ-Плюс, 2010. – 864 с
7. Марк Саммерфилд. Программирование на Python 3. Подробное руководство. – Перевод с английского. – СПб.: Символ-Плюс, 2009.– 608 с.
8. Ноа Гифт, Джереми М. Джонс. Python в системном администрировании UNIX и Linux. – Перевод с английского. – СПб.: Символ-Плюс, 2009. – 512 с
9. Бизли, Дэвид М. Язык программирования Python. Справочник. – К.: ДиаСофт, 2000. – 336 с.
10. Сузи Р. А. Python. Наиболее полное руководство (+CD). – СПб.: БХВ-Петербург, 2002. – 768 с.
11. Сузи Р. А. Язык программирования Python: Учебное пособие. – М.: ИНТУИТ, БИНОМ. Лаборатория знаний, 2006. – 328 с.
12. М. Доусон. Программируем на Python. – СПб.: Питер, 2012. – 432 с.
13. С. Шапошникова. Основы программирования на Python. Учебник. Вводный курс. – версия 2. – 2011. – 44 с.
14. И. А. Хахаев. Практикум по алгоритмизации и программированию на Python. Учебник. — М.: Альт Линукс, 2010. – 126 с.
15. А. Н. Чаплыгин. Учимся программировать вместе с Питоном. Учебник. - 135 с.
16. A. Sweigart. Invent Your Own Computer Games with Python. – 2008 – 2010. – 436 с.
17. A. Sweigart. Core Python Applications Programming. – 2012. – 888 с.