

Тема № 4:

Компютърно моделиране

*Умения за успешно прилагане на учебните програми
по компютърно моделиране и информационни технологии
в прогимназиален етап*

НИКОЛИНА НИКОЛОВА • ЕЛИЗА СТЕФАНОВА • МИРОСЛАВА НИКОЛОВА
• ДИАНА ПЕТРОВА • ОЛЕГ КОНСТАНТИНОВ • СВЕТЛА БОЙЧЕВА • СТАНИСЛАВ ИВАНОВ

ГИМНАЗИАЛЕН ЕТАП

Въведение в ООП
Създаване на приложения с ГПИ
Управление на изчислителния процес
Събития по подразбиране
Масив и символен низ

- Вградени функции
- Библиотеки
- Набори от данни

6. КЛАС

Преминаване от език за блоково програмиране към скриптов текстов език.
Създаване на анимация със средствата на скриптов език



ПРОГИМНАЗИАЛЕН ЕТАП

- Компютърно моделиране и ИТ
- В рамките на 18 уч. ч.
- Подготовка за изучаване на информатика
- Развиване на алгоритмично мислене

7. КЛАС

Използване на скриптов текстов език за реализиране на конкретен проект



5. КЛАС

Блоково програмиране
Циклични процеси
Равнинни фигури
Анимации
Собствени блокове



8. КЛАС

Интегрирани среди за програмиране
Запознаване с език за обектно-ориентирано програмиране
Графичен потребителски интерфейс



ПРОФИЛ

Въведение в концепциите на обектно-ориентираното програмиране
Подготовка за университет или за професионална реализация



Акценти, спираловидно развитие

5. клас

- Създава и използва **собствени блокове** в изучаван език за блоково програмиране
- Създава **образователен проект** със средствата на изучаван език за блоково програмиране
- Прилага блоково програмиране за реализиране на **междупредметни връзки**

6. клас

- Създава **несложна анимация** чрез програмиране на текстов скриптов език с въвеждане на текст.

7. клас

- Определя/избира **типа на данните** съобразно изискванията на конкретна задача
- Прилага подходящи **оператори за цикъл** за реализиране на алгоритъм за решаване на задача
- Реализира **проект в екип** като използва **скриптов текстов език** за програмиране, интернет, различни носители на информация

7. клас

Дигитална култура

- Работа с различни типове данни
- Анализ на проблемна ситуация
- Моделиране на проблемна ситуация

Тема 4: Компютърно моделиране

4.1. Основни типове данни в скриптов текстов език за програмиране

4.2. Приложение на циклични конструкции

Учебно съдържание

Тема 4. Компютърно моделиране

- Основни типове данни в скриптов текстов език за програмиране
 - Изброява примери за реални ситуации, в които се използват различни типове данни
 - Обяснява разликата между различни типове данни и приложението им
 - Разглежда програми, които моделират реални ситуации с използване на различни типове данни
 - Определя типа на данните съобразно изискванията на конкретна задача
- Приложение на циклични конструкции
 - Дава примери за повтарящи се действия
 - Анализира условието на задача, включваща повтарящи се действия
 - Прилага подходящи оператори за цикъл за реализиране на алгоритъм за решаване на задача

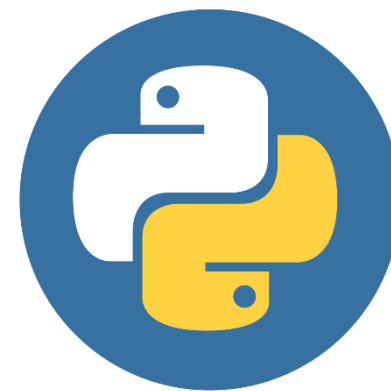
Нови понятия

Понятия	Описание
Числови (цял и реален) тип данни	Именуван тип на данни, който позволява представяне и работа с цели или реални числа
Низове (текстови типове данни)	Именуван тип данни, който позволява работа със символни низове

Реализация

Python

- Надграждане на знанията и уменията от 6. клас
- **Костенуркова** графика
- Вход и изход на данни
- Конвертиране на данни
- Моделиране на по-сложни циклични процеси
- Въвеждане на специфични средства на езика само при необходимост
- Електронен микрокурс в Trinket:
<https://trinket.io/it-schoolbooks-gmail-com/courses/kompyutrno-modelirane-za-7-klas#/>



Учебен пример – въвеждане на ново съдържание



Типове на данните

За решаването на някои задачи са необходими разнообразни входни данни. Тези данни може да бъдат от различни типове – текст, числа, звук и други.

С различните типове данни може да се правят различни неща: да се събират числа, да се отпечата текст и т.н.

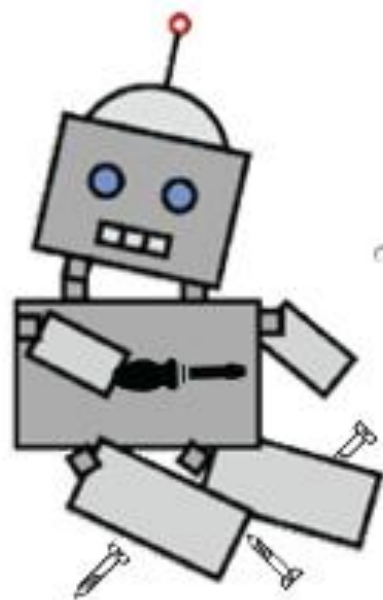
Типът на данните, които се запазват в променливите в Python, определя множеството от стойности, които може да приема променливата, действията, които може да се извършват с нея, както и паметта, която компютърът трябва да отдели за запазването ѝ.

Учебен пример - провокация

```
part_1 = input('Брой болтчета в първа част: ')\npart_2 = input('Брой болтчета във втора част: '\n\ntotal_bolts = part_1 + part_2\nprint('Общо: ' + str(total_bolts))
```



Брой болтчета в първа част: 5
Брой болтчета във втора част: 2
Общо: 52



Ох.. Отново
проблеми.
Помогнете ми!

✓ Проблем 2!

Роботко обича да сглобява различни механизми. Първо подготвя две части, а след това ги свързва. Но, когато изчислява колко болтчета ще му трябват, вечно се обърква. Затова иска да създаде програма, в която да въвежда броя на болтчетата за всяка част и да извежда общия им брой.

След като създаде програма **Bolt.py**, забелязал, че тя не работи правилно. Помогнете му, като откриете и отстраните грешката!

Експериментирайте с кода. Какво забелязвате? На какво според вас се дължи проблемът с кода на *Роботко*?

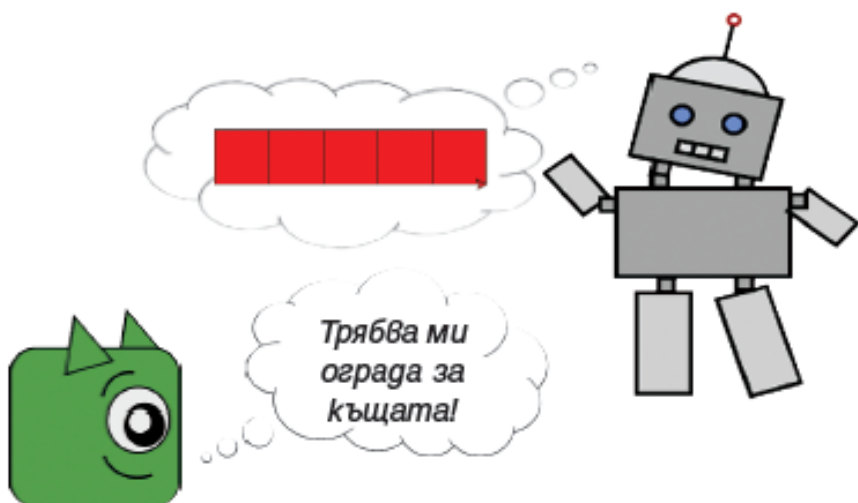
Проблемни ситуации

✓ Мисия

Квадратчо помоли Роботко да построи ограда около къщата му. Преди да започне със строежа, Роботко иска да направи проект на оградата на **Python** и да го покаже на Квадратчо.

Оградата трябва да се състои от 5 долепени една до друга квадратни плоскости с дължина на страната 50.

Помогнете му да създаде проекта!



```
import turtle

robo = turtle.Turtle()

def draw_square(size):
    for i in range(4):
        robo.forward(size)
        robo.left(90)

for square in range(5):
    draw_square(50)
    robo.forward(50)
```

Добавяне на библиотеката *turtle* и създаване на костенурка

Функция за изчертаване на квадрат

Повтарящи се дейности:

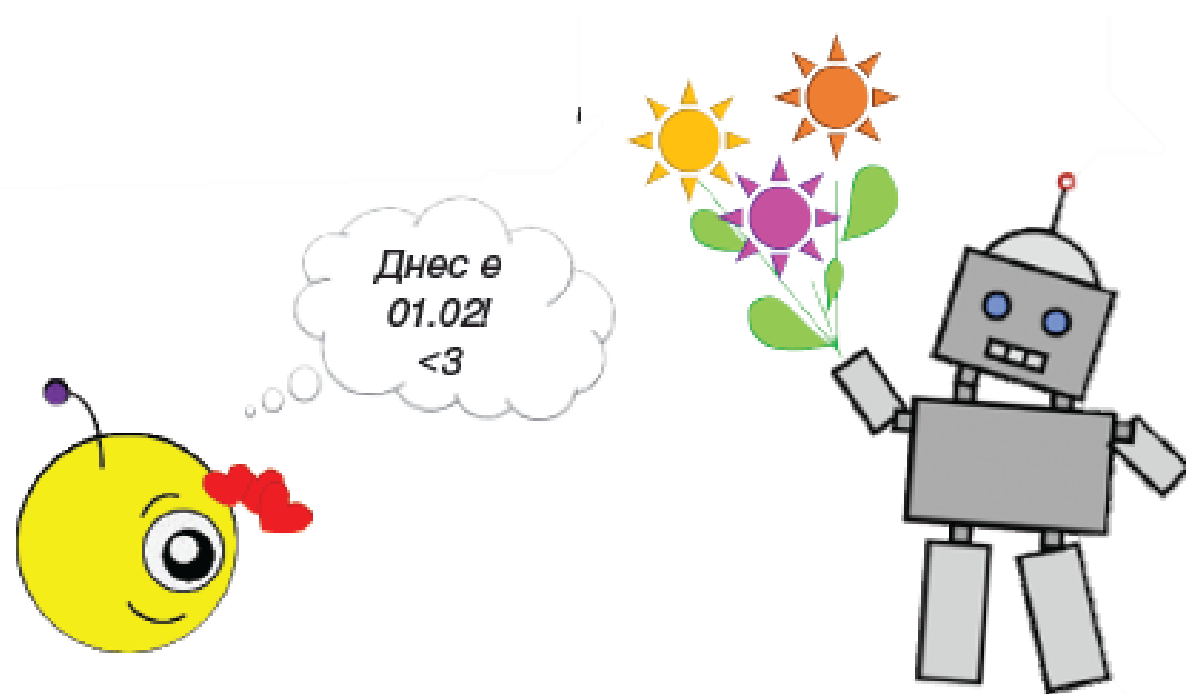
- изчертаване на квадрат
- преместване напред

Допълнителни предизвикателства

✓ Щастieto на Сферичка

Сферичка е суеверна. Тя Вярва, че за Всеки ден има щастливо число, което ѝ помага във Всичко. Щастливото ѝ число се получава лесно – като се съберат цифрите на деня и месеца от текущата дата. Например, за 19.12. щастливото число е $1+9+1+2 = 13$.

Помогнете ѝ да си намира щастливото число за Всеки ден!



Трикове

Трик: Цветна ограда

За да направите цветната ограда, може да използвате списък от цветовете на Сферичка:

1. Създайте цикъл, който обхожда списъка с цветовете на Сферичка.
2. Променете цвета на костенурката robo на всяка стъпка от цикъла на съответния цвят.
3. Изчертайте цветния панел.

```
for color in ["Green", "Blue", "Red", "Yellow", "Purple"]:  
    robo.color("Black", color)
```

```
robo.begin_fill()  
panel(50)  
robo.end_fill()
```



Трик: целочислено деление

В Python има оператори за целочислено деление:

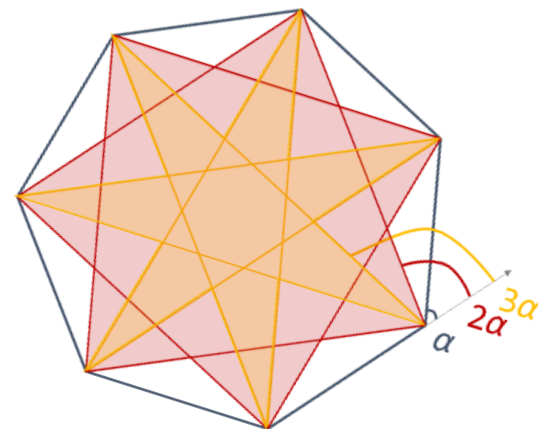
- % – връща остатък от делението на две цели числа.
- // – връща цялата част от делението на две цели числа.

Пример:

```
7 % 3 → 1  
7 // 3 → 2
```

Трик – Ъгъл на въртене за звезда.

Звездата ще има най-остри лъчи, ако ъгълът на въртене е $\frac{n}{2}$ пъти по-голям от ъгъла на завъртане за изчертаване на многоъгълник.



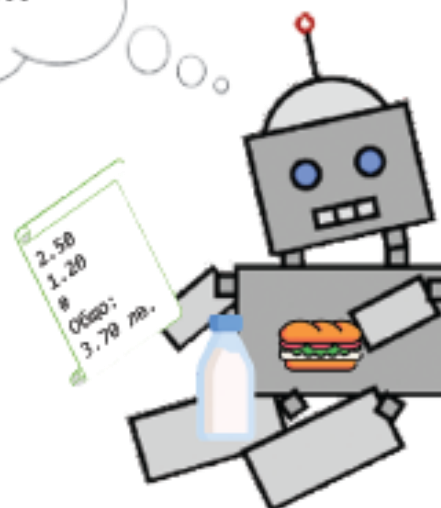
Анализ на проблемни ситуации

Анализ

За да намери общата дължима сума, Роботко трябва:

1. Да въвежда цената на всяка стока.
 2. Веднага след това да я добавя към текущата сума.
- Това означава, че в началото трябва да си **нулира** общата сума, за да не оцети някого неволно.
- Проблемът е, че не знае кой клиент колко стоки ще купи... Ще трябва по някакъв начин да сигнализира на програмата, когато покупките свършат. Той решава за целта да въвежда знака #.
- Какъв оператор за цикъл да използва, за да прекрати въвеждането, когато покупките свършат?!

Как да му кажа да спре?!



Анализ

Данните за количеството фини прахови частици през различните дни може да се запазят в списък. Визуално те може да се представят чрез стълбовидна диаграма с помощта на **Костенурковата графика**. Удобно е за изчертаването на един стълб от диаграмата да се опише функция.

Отново има високо количество фини прахови частици и не мога да излизам.



Програмен код с пояснения

```
# Касов апарат
total = 0
price = input('Цена: ')
while price != '#':
    total = total + float(price)
    print('Текуща сума: ' + str(total))
    price = input('Цена: ')
print('Дължите общо: ' + str(total) + ' лв.')
```

Нулиране на стойността на *total*

Въвеждане на първата цена

Условие,
управляващо цикъла

Повтарящи се дейности:

- Добавяне на цената към общата сума
- Извеждане на текуща сума
- Въвеждане на следваща цена.

```
import turtle

robo = turtle.Turtle()

def draw_square(size):
    for i in range(4):
        robo.forward(size)
        robo.left(90)

for square in range(5):
    draw_square(50)
    robo.forward(50)
```

Добавяне на библиотеката *turtle*
и създаване на костенурка

Функция за изчертаване
на квадрат

Повтарящи се дейности:
• изчертаване на квадрат
• преместване напред

Оператор while: Как се пише?

while условие:

Команди

Команди, които
се повтарят

Условие за извършване
на повторение

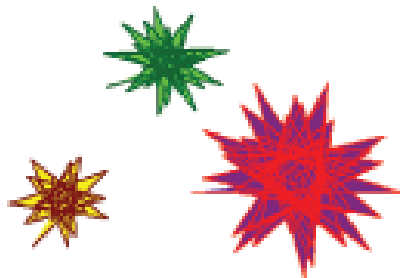
Пример:

```
while price != '#':
    total = total + float(price)
```

Интригуващи проекти

```
def virus(x, y, n, size, contour, filling):  
    robo.goto(x,y)  
    robo.pendown()  
    robo.color(contour, filling)  
    robo.begin_fill()  
    angle = (n//2) * 360 / n - 1  
    while (size > 0):  
        robo.forward(size)  
        robo.left(angle)  
        size = size - 3  
        angle = angle - 1  
    robo.end_fill()  
    robo.penup()
```

Внимание! Вирус!
Спазвайте правилата за хигиена!

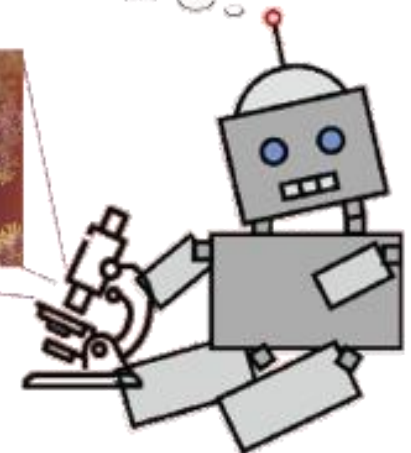
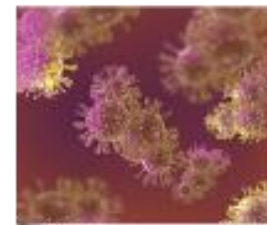


Мисия

Напоследък случаите на вирусни инфекции отново се увеличават. Роботко е социално ангажиран. Решава да направи предупредителен постер с напомняне за спазване на хигиената.

Дали ще успее да моделира вирус, за да илюстрира постера?

Сякаш напомня на звездите, които рисувах миналата година...



И още много в нашето студио: <https://trinket.io/it-schoolbooks-gmail-com/courses/kompyutrno-modelirane-za-7-klas#/prilozhenie-na-ciklichni-konstrukcii/prilozhenie-na-ciklichni-konstrukcii>

Методически белешки

Препоръки

- Първата част:
 - Фокусът върху работата с променливи и основните типове данни – целочислен, реален, низ
 - При създаването на програми, моделиращи реални ситуации с използването на различни типове данни – анализ на входните и изходните данни на програмата чрез подходящи въпроси
- Втора част
 - Приложение на циклични конструкции
 - Надграждане след 6. клас: Костенуркова графика
 - Акцент: приложението на цикличните конструкции за решаване на конкретна задача
 - избор на циклична конструкция
 - създаване и описание на цикличен алгоритъм
 - Подготовка за гимназиален етап с акцентиране върху:
 - използваните библиотеки – `turtle`, `random`
 - вградени функции – `print()`, `zip()`
 - типове данни и структури – `int`, `float`, списък

Предизвикателства и възможности

Предизвикателства

- Математически апарат
- Техника и софтуер
- Време за работа
 - Заготовки
 - Междинни решения
 - Демонстрации
 - Проследяване на код

Възможности

- **Диференциация** – учениците с повече опит може да
 - Проучват самостоятелно нови команди
 - Подпомагат по-бавните
 - Експериментират
- **Творчество**
 - Подход към решението
 - Краен резултат
 - Реализация на собствени идеи
- **Интегриране в проектна дейност**
 - Междинни изчисления
 - Графика

Ресурси

- Ресурсите за учителя: <https://www.e-prosveta.bg/>
- В Trinket курса на издателството: <https://trinket.io/it-schoolbooks-gmail-com/courses/kompyutrno-modelirane-za-7-klas#>

Готови ли сте за следващата тема?