Дефиниране на класове

Абстрактни типове данни и дефиниране на класове

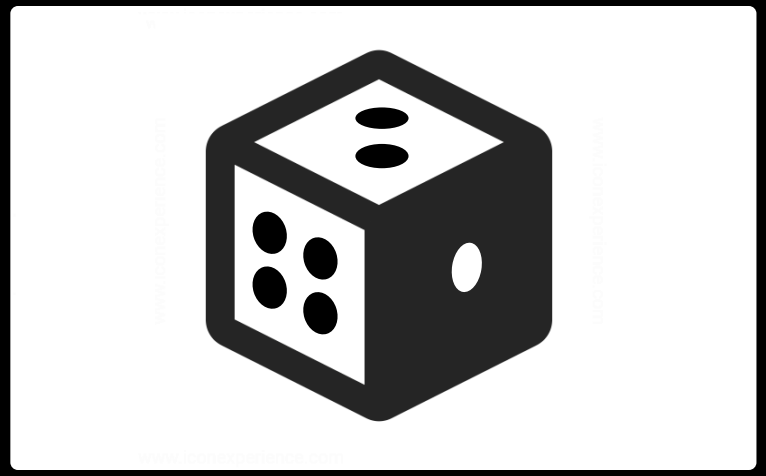
Учителски екип

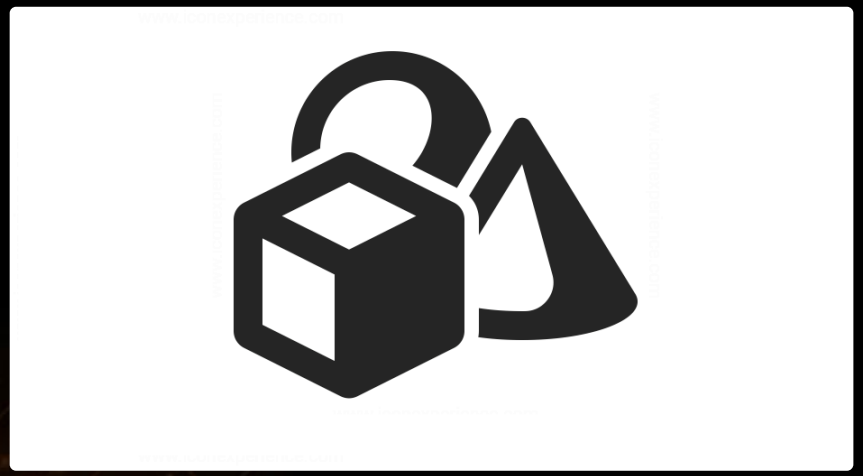
Обучение за ИТ кариера

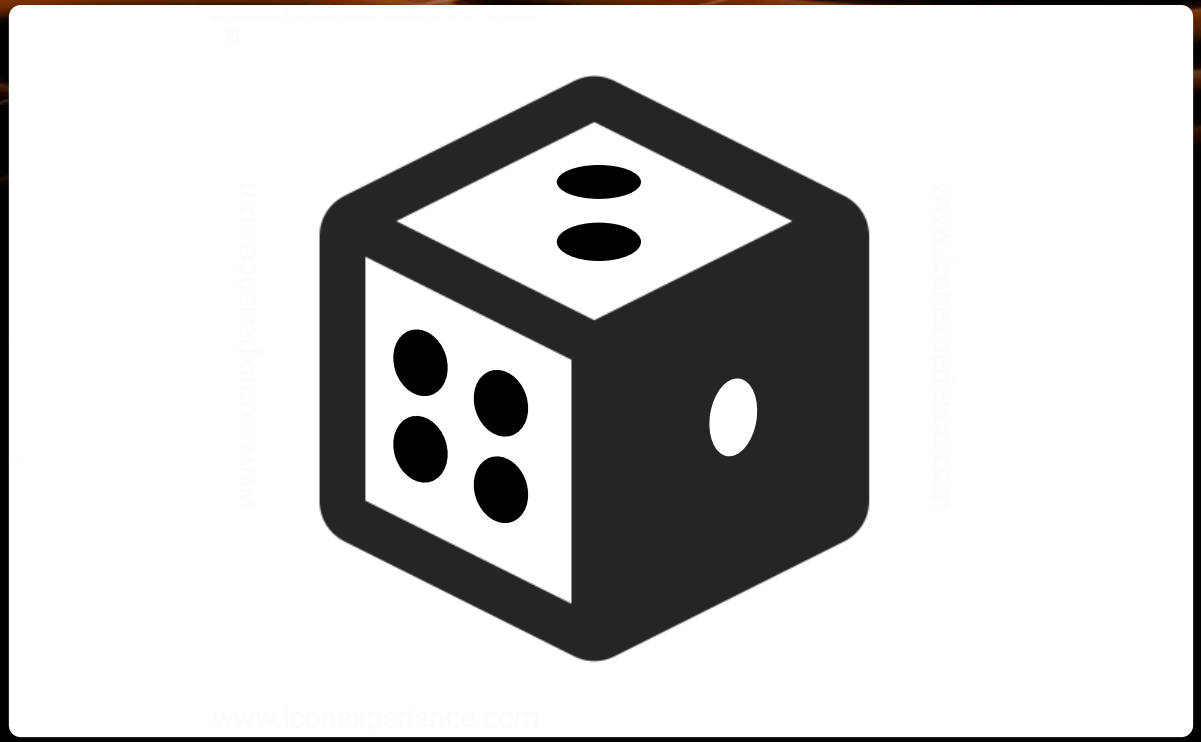
https://it-kariera.mon.bg/e-learning/

Съдържание

1. Абстрактни типове данни
2. Класове
3. Обекти







Абстрактни типове данни

Скриване на детайлите

Абстрактни типове данни

- Абстрактните типове данни описват:
  - Множество от данни
  - Възможни операции в рамките на този тип

  Абстрактните типове данни (АТД) ни позволяват да опишем конкретна структура (т.е. нейните данни и операции), без обаче да се интересуваме от детайлите в тази реализация

Абстрактни типове данни

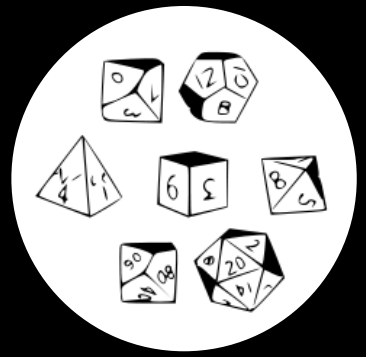
- Не е нужно да знаем как нещо е направено, за да ползваме АТД

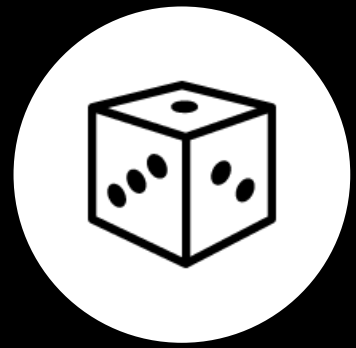
```
Dog:
          Dog()
 string Name()
    void Bark()
    void Sleep()
```

```
Computer:
          Computer()
    void TurnOn()
    void TurnOff()
 string Spec()
```

Класове и Обекти

- Класовете служат за създаване на „имплементация“ на АТД
- Класовете ни позволяват да описваме и създаваме обекти
- Обектът е една инстанция на класа

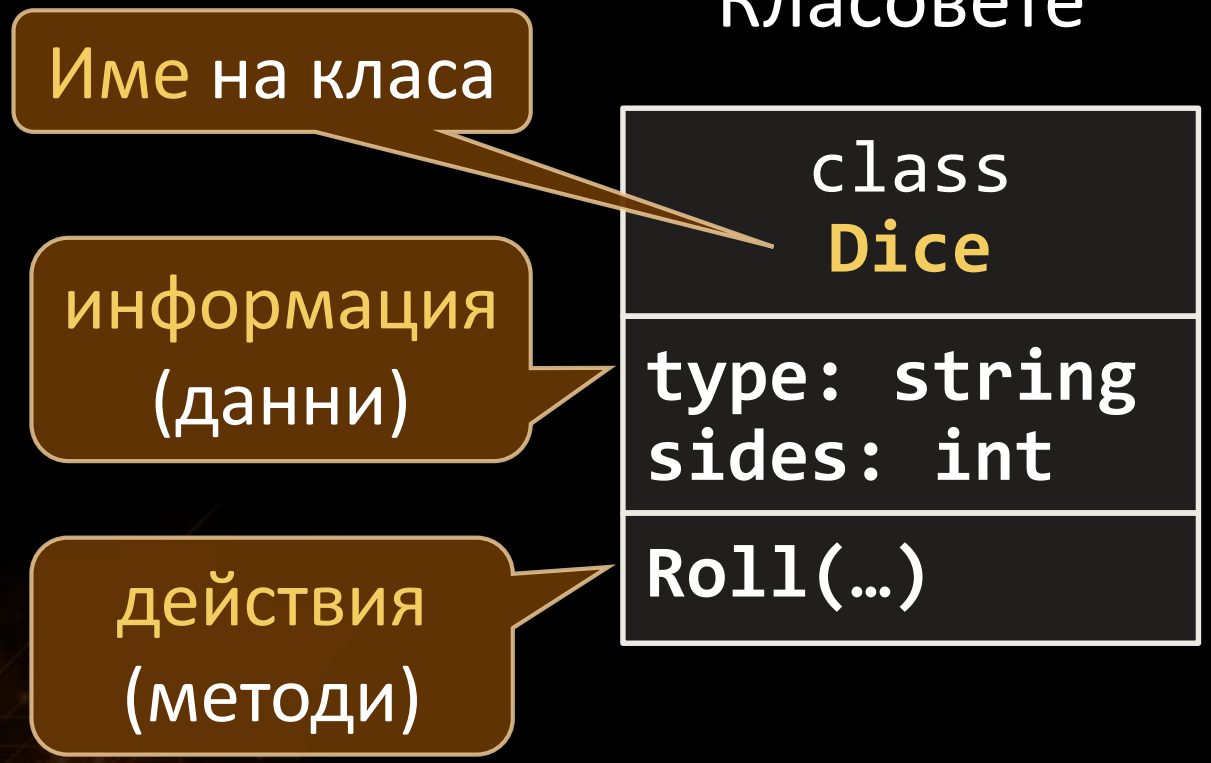




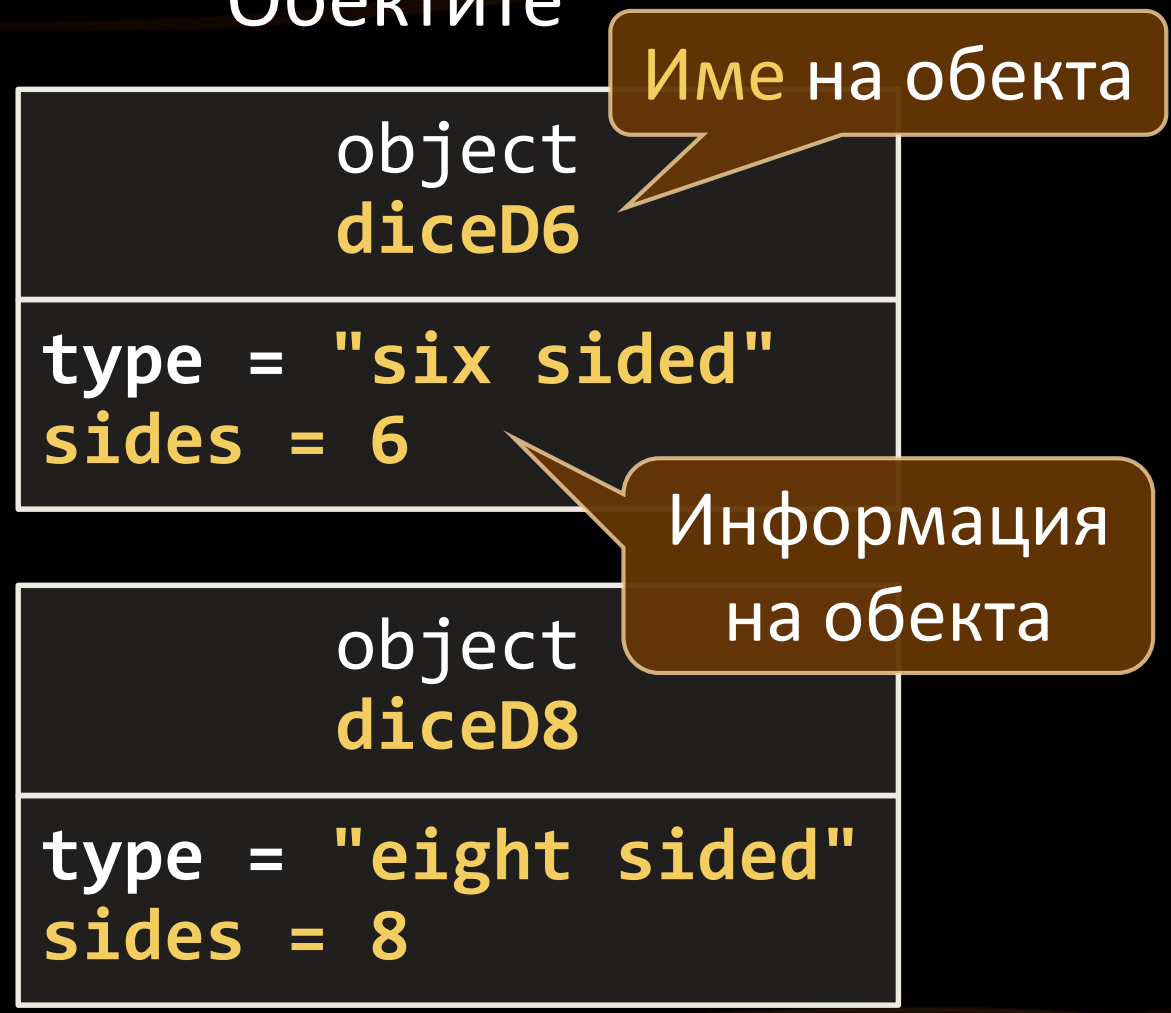
АТД Зар → Зар (Class) → Зар с 6 страни (Обект)

Класове и Обекти (2)

Класовете



Обектите



Дефиниране на прост клас

- Вие вече сте ползвали класове - например класа Program
- Можем по същия начин да дефинираме и други класове:

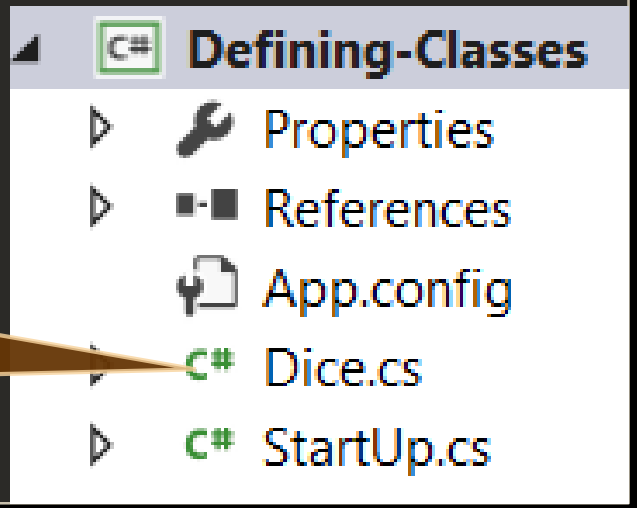
Ключова дума

Име на класа

```
class Dice
{
    ...
}
```

Тяло на клас

Клас в отделен файл



Членове на класа

- Класа съдържа състояния и действия
- Полетата съдържат състоянието
- Методите описват действията

```
class Dice {
  int sides;
  string type;

  void Roll(){ … }
}
```

Полета

Методи

Създаване на обект

- Класът може да има много инстанции (обекти)

```
class Program {
  public static void Main() {
    Dice diceD6 = new Dice();
    Dice diceD8 = new Dice();
  }
}
```

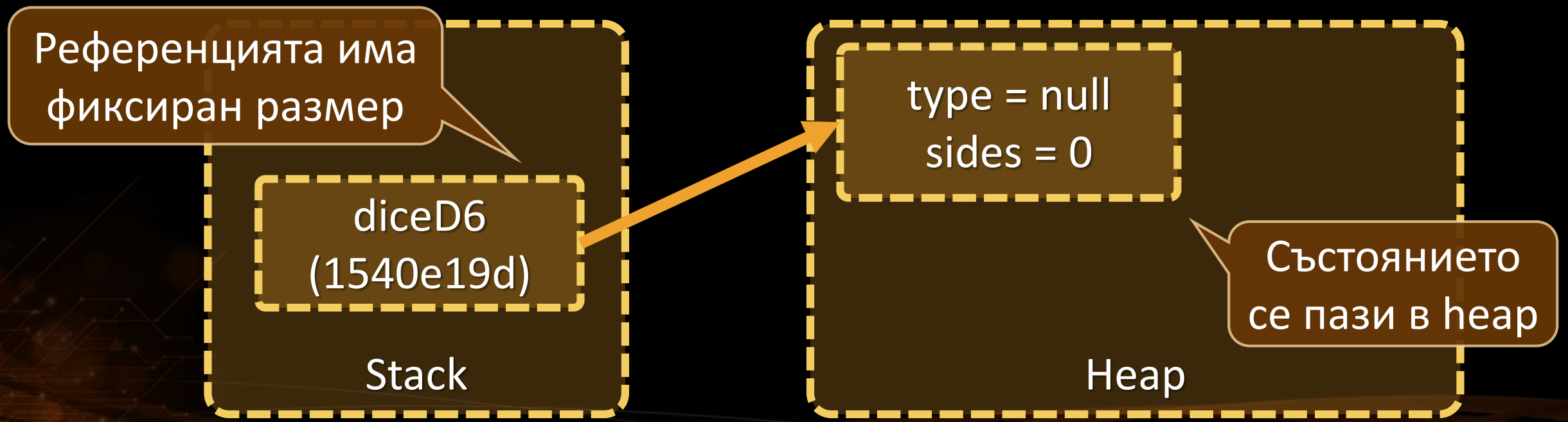
Променливите съдържат **референции**

Използвайте **new**

Обектна референция

- Декларирането на променлива създава референция
- **new** заделя място за данните на обекта в Heap паметта

```
Dice diceD6 = new Dice();
```

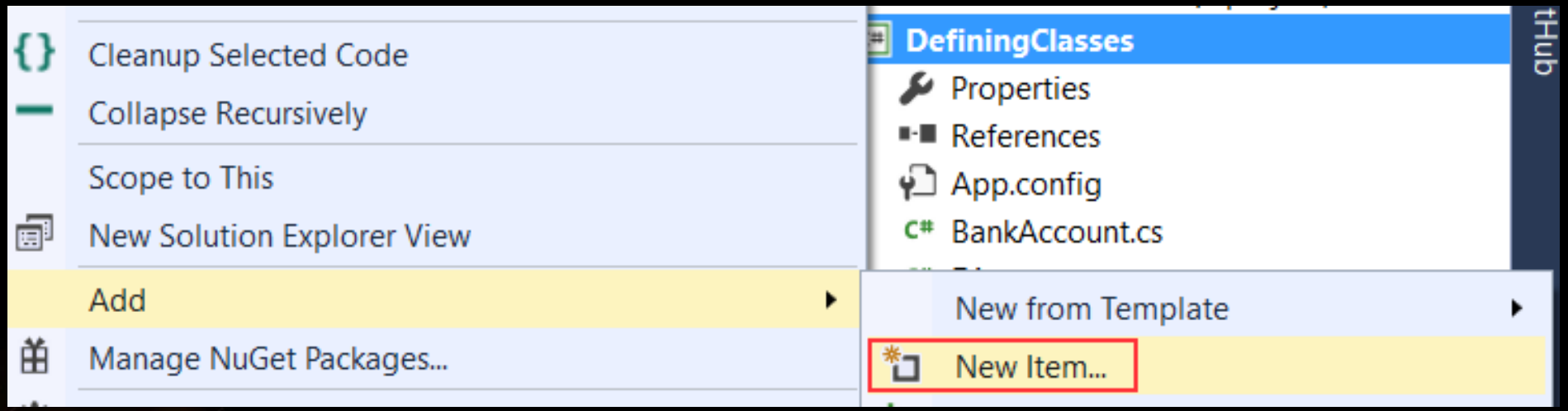


Задача: Дефинирайте клас Person

- Дефинирайте клас `Person`, като за него пазете информация за името и възрастта на човек и реализирайте единствено действието `IntroduceYourself()`, което отпечатва представяне на човека.
- След това създайте и използвайте обект от класа `Person`.

Решение: Дефинирайте клас Person (1)

- Нека първо да създадем файл за този клас:
  [Project] → [Add Class] или:
  десен бутон върху проекта [Add]→ [New Item] → [Class]

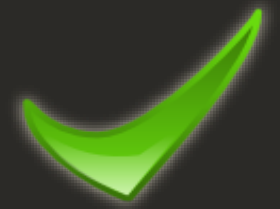


- Внимавайте с именуването на класа

Именуване на класове

- Класовете са PascalCase
- Използвайте описателни съществителни
- Избягвайте абревиатури (освен известни: URL, HTTP, и др.)

```
class Dice { … }
class BankAccount { … }
class IntegerCalculator { … }
```



```
class TPMF { … }
class bankaccount { … }
class intcalc { … }
```

Решение: Дефинирайте клас Person (2)

```
class Person {
  public string name;
  public int age;

  public void IntroduceYourself() {
    Console.WriteLine("Здравейте! Аз съм" +
      " {0} и съм на {1} години.", name, age);
  }
}
```

Решение: Дефинирайте клас Person (3)

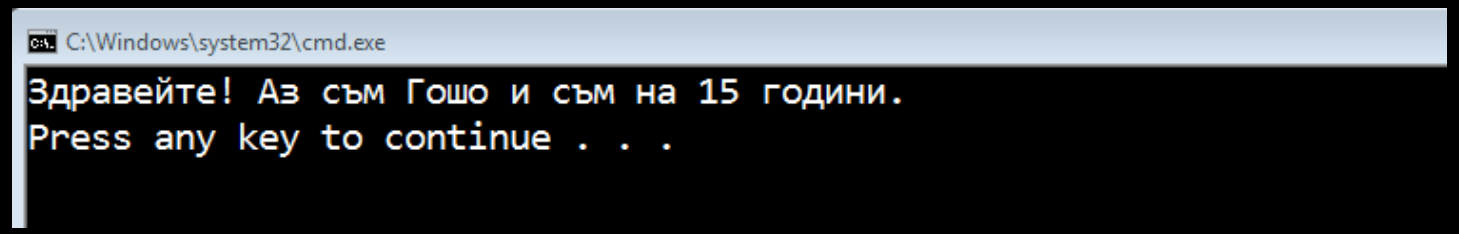
- Сега е време да използваме класа и да създадем обект в Main метода в Program.cs:

```
static void Main(string[] args) {
  Person firstPerson = new Person();
  firstPerson.name = "Гошо";
  firstPerson.age = 15;

  firstPerson.IntroduceYourself();
}
```

Решение: Дефинирайте клас Person (4)

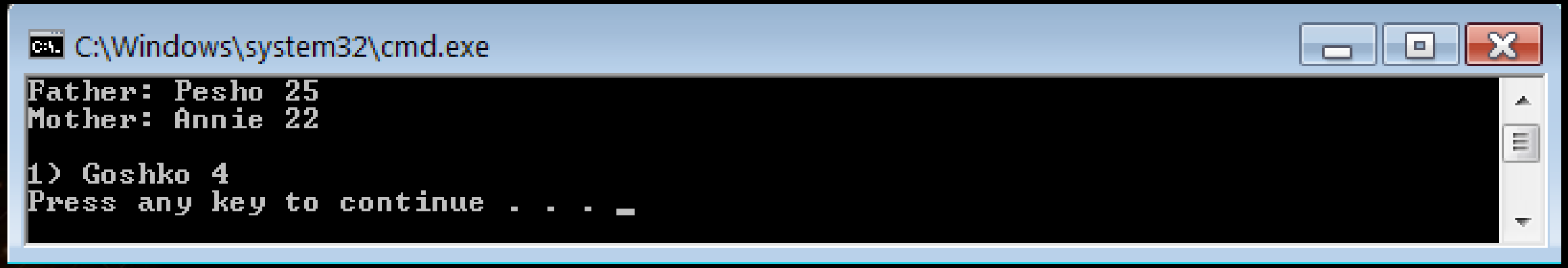
- Ако сте работили правилно ще получите:



- Аналогично създайте `secondPerson` и `thirdPerson` и извикайте `IntroduceYourself` и за тях

Задача: Семейство

- Дефинирайте клас `Family`. В него пазете информация за майката и бащата (от тип `Person`) и списък на техните деца
- Направете метод `IntroduceFamily()`, който извежда информацията за това семейство.



Решение: Семейство(1)

- Първо ще създадем файл за клас `Person`.
- След това ще направим клас Family, който в полета от тип `Person` ще съхранява данните за майката и бащата.
- Децата може да са различен брой, ще ги пазим в обект от тип списък (List) от хора (Person)
- После в класа `Person` ще добавим метод, който извежда данните за един човек
- Накрая в класа Family ще добавим метод, който извежда данните за цялото семейство

Решение: Семейство (2)

- В `Person.cs` описваме данните за един човек и как става извеждането им:

```
public class Person
{
    //TODO: добавете полета за име и възраст

    public void IntroduceMyself(string text)
    {
        Console.WriteLine(text + name + " " + age);
    }
}
```

Решение: Семейство (3)

- Във `Family.cs` описваме от какво се състои семейството:

```
class Family
{
    public Person father = new Person();
    public Person mother = new Person();

    public List<Person> children = new List<Person>();

    public void IntroduceFamily() {...}
}
```

Решение: Семейство (4)

- … и как се извежда тази информацията:

```
public void IntroduceFamily()
{
    father.IntroduceMyself("Father: ");
    mother.IntroduceMyself("Mother: ");
    Console.WriteLine();
    for(int i = 0; i < children.Count; i++)
    {
        children[i].IntroduceMyself((i + 1) + ") ");
    }
}
```

Решение: Семейство (5)

- Накрая в `Program.cs` попълваме данните за семейството:

```
Family myFamily = new Family();

myFamily.father.name = "Pesho";
myFamily.father.age = 25;

// TODO попълнете данните за майката

// TODO създайте обект за детето и попълнете данните му
myFamily.children.Add(child);

myFamily.IntroduceFamily();
```

Какво научихме?

- Абстрактните типове данни:
  - Описват нещо чрез неговите данни и възможните действия с тях
- Класовете описват конкретна реализация на АТД
  - При дефинирането на клас се описват неговите полета, методи и други членове
- Обектите са инстанция на класа
  - При създаването на обект променливата съдържа само референция към обекта

Абстрактни типове данни и класове

Въпроси?

https://it-kariera.mon.bg/e-learning/

Лиценз

- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"

- Благодарности: настоящият материал може да съдържа части от следните източници
  - Книга "Основи на програмирането със C#" от Светлин Наков и колектив с лиценз CC-BY-SA