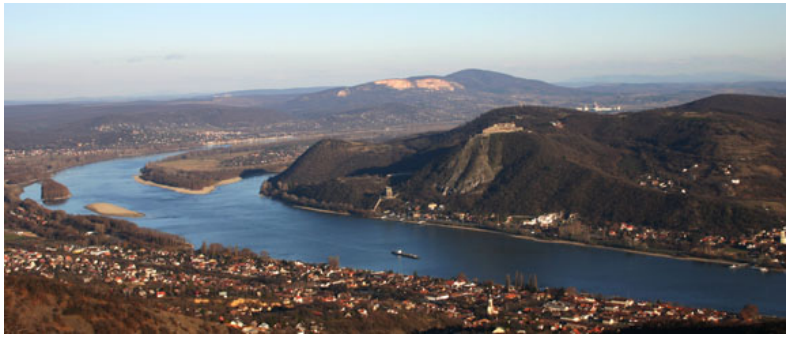
Класове, методи, this

Калин Георгиев

27 февруари 2018 г.

Моделиране. Абстракция със структури от данни



```
struct River
{
  char name[100];
  double waterLevels[365];
}
```

```
struct Person
{
  char name[100];
  Date birthdate;
}
```

Логическа сигнатура

$$< \mathcal{D}; f_1, f_2, \ldots, f_k; p_1, p_2, \ldots, p_l >$$

- Носител (множество допустими стойности) – $\mathcal{D}$
- Функции (операции) – $f : \mathcal{D}^n \to \mathcal{D}$
- Предикати – $p : \mathcal{D}^n \to \{tt, ff\}$

Пример: Множество от букви

$$< 2^{\{'a'..'z'\}}; \cup, \cap; \mathit{empty} >$$

- $A \cup B = \{x | x \in A || x \in B\}$
- $A \cap B = \{x | x \in A \&\& x \in B\}$
- $\mathit{empty}(A) = \begin{cases} \mathit{ff} & \text{if } \exists x \in A \\ \mathit{tt} & \mathit{otherwise} \end{cases}$

Съответна СД

```
struct CharSet
{
  bool contents[26];
};
```

$x \in A \Leftrightarrow$

```
A.contents[x-'a'] == true
```

Съответна СД

```
struct CharSet
{
  bool contents[26];
};

bool empty (CharSet s)
{
  for (int i = 0; i < 26; i++)
    if (s.contents[i])
      return false;
  return true;
}
??? setUnion (????,????)
{
  ....
}
```

Съответна СД

```
struct CharSet
{
  bool contents[26];
};
bool empty (CharSet s)
{
  for (int i = 0; i < 26; i++)
    if (s.contents[i])
      return false;
  return true;
}
CharSet setUnion (CharSet a, CharSet b)
{
  CharSet result;
  for (int i = 0; i < 26; i++)
    result.contents[i] = a.contents[i] || b.contents[i];
  return result;
}
```

Как да обединим данните и операциите: CLASS

Class

```
class CharSet
{
  public:
  bool contents[26];
  bool empty () //!!!
  {
    for (int i = 0; i < 26; i++)
      if (contents[i])
        return false;
    return true;
  }
  CharSet setUnion (CharSet b)
  {
    CharSet result;
    for (int i = 0; i < 26; i++)
      result.contents[i] = contents[i] || b.contents[i];
    return result;
  }
}
```

class vs. struct

```
class CharSet
{
  public:
  bool contents[26];
  bool empty () //!!!
  {
    for (int i = 0; i < 26; i++)
      if (contents[i])
        return false;
    return true;
  }
  CharSet setUnion (CharSet b)
  {
    CharSet result;
    for (int i = 0; i < 26; i++)
      result.contents[i] =
        contents[i] || b.contents[i];
    return result;
  }
}
```

```
struct CharSet
{
  bool contents[26];
};
bool empty (CharSet s)
{
  for (int i = 0; i < 26; i++)
    if (s.contents[i])
      return false;
  return true;
}
CharSet setUnion (CharSet a, CharSet b)
{
  CharSet result;
  for (int i = 0; i < 26; i++)
    result.contents[i] =
      a.contents[i] || b.contents[i];
  return result;
}
```

Класове / обекти

```
int main ()
{
  CharSet s1,s2,s3;

  //initialization

  s3 = s1.setUnion (s2);

}
```

```
class CharSet
{
  public:
  bool contents[26];
  bool empty () //!!!
  {
    for (int i = 0; i < 26; i++)
      if (contents[i])
        return false;
    return true;
  }
  CharSet setUnion (CharSet b)
  {
    CharSet result;
    for (int i = 0; i < 26; i++)
      result.contents[i] =
        contents[i] || b.contents[i];
    return result;
  }
}
```

this: CharSet*

this

```
int main ()
{
  CharSet s1,s2,s3;

  //initialization

  s3 = s1.setUnion (s2);

}
```

```
class CharSet
{
  public:
  bool contents[26];
  bool empty () //!!!
  {
    for (int i = 0; i < 26; i++)
      if (this->contents[i])
        return false;
    return true;
  }
  CharSet setUnion (CharSet b)
  {
    CharSet result;
    for (int i = 0; i < 26; i++)
      result.contents[i] =
        this->contents[i] || b.contents[i];
    return result;
  }
}
```

Прости оператори

Прости оператори

$$c = a.setUnion\ (b) \sim c = setUnion\ (a,b)$$

$$c = a + b$$

Прости оператори

$$c = a.setUnion\ (b) \sim c = setUnion\ (a,b)$$

$$c = a + b$$

Прости оператори

```
class CharSet
{
//....
  CharSet setUnion (CharSet b){...}
};
```

```
class CharSet
{
//....
  CharSet operator + (CharSet b){...}
};
```

Прости оператори

```
class CharSet
{
//....
  CharSet setUnion (CharSet b){...}
};
```

```
class CharSet
{
//....
  CharSet operator + (CharSet b){...}
};
```

Благодаря за вниманието!