

OBJEKTNO ORIENTIRANO PROGRAMIRANJE

1

KAJ JE RAZRED?
KAJ JE OBJEKT?
KAKO USTVARIMO OBJEKTE?
KAKO UPORABLJAMO OBJEKTE?

Kako smo programirali do sedaj?

2

- strukturirano / proceduralno programiranje
 - imamo nabor podatkovnih tipov
 - celoten program razdelimo na manjše celote – funkcije
 - funkcije lahko združimo v logične celote – knjižnice (module)
 - funkcije izvajamo – kličemo – na naslednji način:
 - × $f(x)$ – funkcijo f izvedi nad podatkom x
 - poudarek je nekako na postopkih, ki obdelujejo podatke

Objektno orientirano programiranje - OOP

3

- objektno orientirano programiranje ni novo
- gre za nadgradnjo strukturiranega programiranja
- podatkovne tipe lahko razvrstimo v hierarhijo
 - obstoječe podatkovne tipe nadgradimo
 - to nam omogoča dedovanje
- prednosti OOP
 - lažje vzdrževanje kode
 - lažje spreminjanje kode
 - ponovna uporabnost kode

OOP – ključne ideje in lastnosti

4

- osnovna ideja OOP:
 - z objekti modeliramo stanje iz realnega sveta
 - nalogo rešimo z objekti, ki si pošiljajo sporočila
- ključne lastnosti OOP
 - OGRAJEVANJE / KAPSULACIJA
 - × objekti ograjujejo (združujejo, kapsulirajo) stanja in obnašanja
 - DEDOVANJE
 - × omogoča nadgrajevanje obstoječih podatkovnih tipov
 - × tako je olajšano vzdrževanje in spreminjanje programov
 - POLIMORFIZEM
 - × različni objekti se za isto obnašanje odzovejo različno
 - SKRIVANJE PODATKOV
 - × uporabniku omogočamo dostop do stanj preko funkcij
 - × zagotavljamo integriteto objekta

Kaj je objekt? Kaj je razred?

5

- **OBJEKT**
 - je skupek lastnosti in obnašanj
 - lastnosti
 - × opisujemo stanja objekta
 - × predstavimo jih s spremenljivkami / drugimi objekti
 - obnašanja
 - × dostopajo do oz. spreminjajo lastnosti objekta
 - × predstavimo jih s funkcijami
- **RAZRED**
 - je načrt, ki opisuje objekt določene vrste
 - je model, po katerem tvorimo objekte
 - objekt je primerek nekega razreda

Primeri opisa objektov (1)

6

- Ko opisujemo objekte (načrtujemo razred) določimo:

1. LASTNOSTI in
2. OBNAŠANJA

PRIMERI:

objekt	lastnosti	obnašanja
pes	barva dlaka pasma ime	laja grize spi je
TV	znamka diagonala trenutni kanal število kanalov	prestavi kanal nastavi barvo išči kanal

Primeri opisa objektov (2)

7

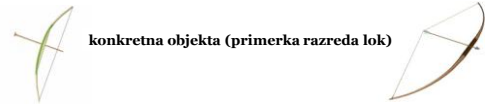
objekt	lastnosti	obnašanja
točka v ravnini	x koordinata y koordinata	premakni točko zrcali preko ... rotiraj okoli druge točke
premica v ravnini • $y=k*x+n$ • $a*x+b*y+c=0$ • dve točki v ravnini	več možnosti: a) k, n b) a, b, c c) dve različni točki v ravnini	premik premice zrcaljenje preko ... rotacija okoli druge točke
kompleksna števila $a+bi$	a, b	seštej odštej zmnoži deli konjugiraj

predstavitve lastnosti je neodvisna od obnašanj (primer premice)

Primer razreda Lok

8

splošen opis objekta lok		
objekt	lastnosti	obnašanja
lok	barva število puščic je napet?	napni sproži



konkretna objekta (primerka razreda lok)

objekt	lastnosti	objekt	lastnosti
mojLok	barva = zelena število puščic = 5 je napet? = false	tvojLok	barva = rjava število puščic = 1 je napet? = true

Deklaracija razreda v C++

9

- razred v C++ običajno zapišemo v dve datoteki
 - deklaracija razreda (.h)
 - definicija razreda (.cpp)
- v deklaraciji samo opišemo razred
 - opišemo njegove lastnosti – razredne spremenljivke
 - opišemo njegova obnašanja – razredne funkcije
- v definiciji definiramo obnašanja
 - definiramo vse napovedane razredne funkcije

Deklaracija razreda v C++

10

- sintaksa za deklaracijo razreda


```
class ImeRazreda {
    // določilo dostopa
    // seznam lastnosti in obnašanj
    ...
};
```
- seznam lastnosti so deklaracije spremenljivk
 - spremenljivk v deklaraciji razreda NE SMEMO inicializirati
- seznam obnašanj so funkcije
 - funkcijo lahko na tem mestu samo deklariramo (napovemo), implementiramo pa kasneje (tako bomo delali mi)
 - funkcijo lahko v celoti definiramo tukaj

Deklaracija razreda v C++: primer Lok
datoteka: Lok.h

11

```
class Clok {
public:
    string barva;
    int stPuscic;
    bool jeNapet;

    bool napni(); // vrne true, če je lok uspešno napet
    int sprozi(); // vrne št. pik, ki smo jih zadeli
};
```

deklaracije lastnosti – so spremenljivke

deklaracije obnašanj – so funkcije
tukaj funkcije samo napovemo, definirali jih bomo izven deklaracije razreda

Definicija razrednih funkcij v C++

12

- funkcijo definiramo na podoben kot do sedaj

```
tip ImeRazreda::imeFunkcije (parametri) {
    ...
}
```

s tem povemo, v kateri razred spada funkcija
funkcija mora biti napovedana v razredu

operator dostopa

- vse napovedane funkcije morajo biti tudi definirane!!

Definicija razrednih funkcij v C++: primer Lok
datoteka: Lok.cpp

13

```
bool CLok::napni() {
    if (stPuscic > 0)
        jeNapet = true;
    else
        jeNapet = false;
    return jeNapet;
}
```

Definicija razrednih funkcij v C++: primer Lok
datoteka: Lok.cpp

14

```
int CLok::sprozi() {
    if (!jeNapet) {
        cout << "Ne morem sproziti. Lok ni napet!";
        return 0;
    }
    int x = rand()% 11; // zadetek na tarci
    stPuscic--;
    jeNapet = false;
    return x;
}
```

Ustvarjanje in uporaba objektov v C++

15

- spremenljivki, katere tip je razred, pravimo **objekt**
- spremenljivko razrednega tipa Razred ustvarimo na naslednji način:

Razred objekt;

- do razrednih spremenljivk in funkcij danega objekta dostopamo preko operatorja . (pika)
 - objekt.imeSpremenljivke
 - objekt.imeFunkcije(...)
- če je seveda dovoljen dostop (več o tem v nadaljevanju)

Ustvarjanje objektov v C++
primer uporabe razreda CLok

16

```
#include "Lok.h"
...
int main() {
    ...
    CLok mojLok;
    mojLok.barva = "bela";
    mojLok.stPuscic = 5;
    if (mojLok.napni()) {
        int zadetek = mojLok.sprozi();
        cout << "Dobil(a) si " << zadetek;
        cout << " tock." << endl;
    }
    ...
}
```

povedati moramo, kje je deklariran razred CLok

ustvarimo objekt *mojLok*, ki je tipa CLok

objektu *mojLok* spremenimo vrednost spremenljivke barva

objektu *mojLok* sporočimo, da izvede metodo *sprozi()*

Pomembna opomba

17

- vsak objekt nekega razreda ima praviloma svoje vrednosti za razredne spremenljivke
- obstajajo izjeme – o njih kasneje

```
int main() {
    ...
    CLok mojLok, tvojLok;
    mojLok.barva="zelena";
    mojLok.stPuscic = 5;
    tvojLok.barva="rjava";
    tvojLok.stPuscic = 1;
    ...
}
```

```
class CLok {
public:
    string barva;
    int stPuscic;
    bool jeNapet;

    bool napni();
    int sprozi();
};
```

mojLok	tvojLok
barva = zelena	barva = rjava
stPuscic = 5	stPuscic = 1
jeNapet = ???	jeNapet = ???