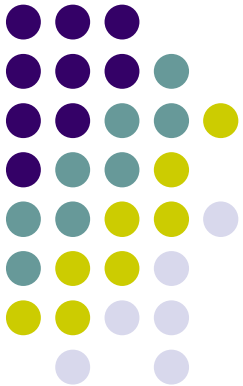
Příklady

doc. Ing. Miroslav Beneš, Ph.D.
katedra informatiky FEI VŠB-TUO
A-1007 / 597 324 213
http://www.cs.vsb.cz/benes
Miroslav.Benes@vsb.cz



Vkládání do stromu – nerekurzivní varianta

```
public boolean insert(Comparable key)
{
  Node node = root; // aktuální uzel
  Node last = null; // předchozí uzel
  int c = 0;        // výsledek porovnání

  while( node != null ) {
    c = key.compareTo(node.getData());
    if( c == 0 ) return false;
    last = node;
    if( c < 0 ) node = node.getLeft();
    else        node = node.getRight();
  }
```

Vkládání do stromu – nerekurzivní varianta

```
  node = new NodeImpl(key, null, null);
  if( root == null )
    root = node;
  else if( c < 0 )
    last.setLeft(node);
  else
    last.setRight(node);
  return true;
}
```

Vkládání do stromu - rekurzivní varianta

```
public void insertRec(Comparable key) {
  root = insertRec(root, key);
}

private Node insertRec(Node node, Comparable key) {
  if( node == null )
    return new NodeImpl(key, null, null);
  int c = key.compareTo(node.getData());
  if( c < 0 )
    node.setLeft(insertRec(node.getLeft(), key));
  else if( c > 0 )
    node.setRight(insertRec(node.getRight(), key));
  return node;
}
```

Soubor s přímým přístupem

```
import java.io.RandomAccessFile;

RandomAccessFile file;
try {
  file = new RandomAccessFile(
           "data.bin",  // jméno souboru
           "rw");       // režim
  …
} finally {
  file.close();
}
```

Soubor s přímým přístupem

```
// velikost souboru
long len = length();
// zjištění pozice v souboru
long pos = getFilePointer();
// nastavení pozice v souboru
void seek(long pos);
// nastavení na začátek souboru
file.seek(0);
// nastavení na konec souboru (append)
file.seek(file.length());
```

Čtení dat
ze souboru s přímým přístupem

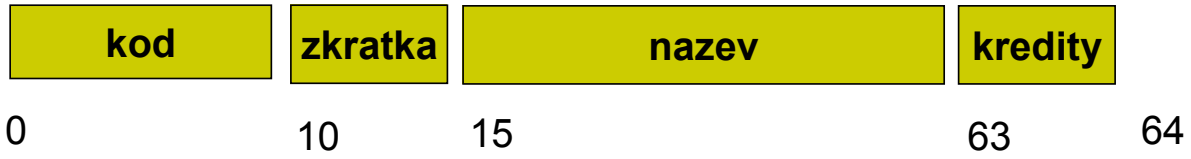
- int read() // 1 byte
- int read(byte[]) // posloupnost bytů
  int readFully(byte[] b)
- boolean readBoolean() // vnitřní formát!
  char readChar()
  double readDouble()
  int readInt()
  …
  String readUTF()

Zápis dat
do souboru s přímým přístupem

- void write(int b)
  void write(byte[] b)
- void writeBoolean(boolean v)
  void writeInt(int v)
  void writeChar(char v)
  void writeChars(String v)
  …
  void writeUTF(String v)

Příklad – seznam předmětů

class Predmet:
    kod: “456-522/1”
    zkratka: “UPR”
    nazev: “Úvod do programování”
    kredity: 6



64 B / záznam

Zápis řetězce pevné šířky

```
void writeString(RandomAccessFile file,
                 String s, int width)
  throws IOException
{
  byte[] data = s.getBytes();
  for(int i = 0; i < width; i++)
  {
    if( i < data.length )
      file.write(data[i]);
    else
      file.write(' ');
  }
}
```

Zápis záznamu o předmětu

```
static final int DELKA_KOD = 10;
static final int DELKA_ZKRATKA = 5;
static final int DELKA_NAZEV = 48;

public void write(RandomAccessFile file)
  throws IOException
{
  writeString(file, kod, DELKA_KOD);
  writeString(file, zkratka, DELKA_ZKRATKA);
  writeString(file, nazev, DELKA_NAZEV);
  file.writeByte(kredity);
}
```

Čtení řetězce pevné šířky

```
String readString(RandomAccessFile file,
                  int width)
  throws IOException
{
  byte[] data = new byte[width];

  int len = file.read(data);

  if( len < 0 )
    return null;

  return new String(data).trim();
}
```

Čtení záznamu o předmětu

```
static Predmet read(RandomAccessFile file)
  throws IOException
{
  String kod = readString(file, DELKA_KOD);
  if( kod == null ) return null;

  Predmet predmet = new Predmet();
  predmet.setKod(kod);
  predmet.setZkratka(readString(file, DELKA_ZKRATKA));
  predmet.setNazev(readString(file, DELKA_NAZEV));
  predmet.setKredity(file.readByte());
  return predmet;
}
```

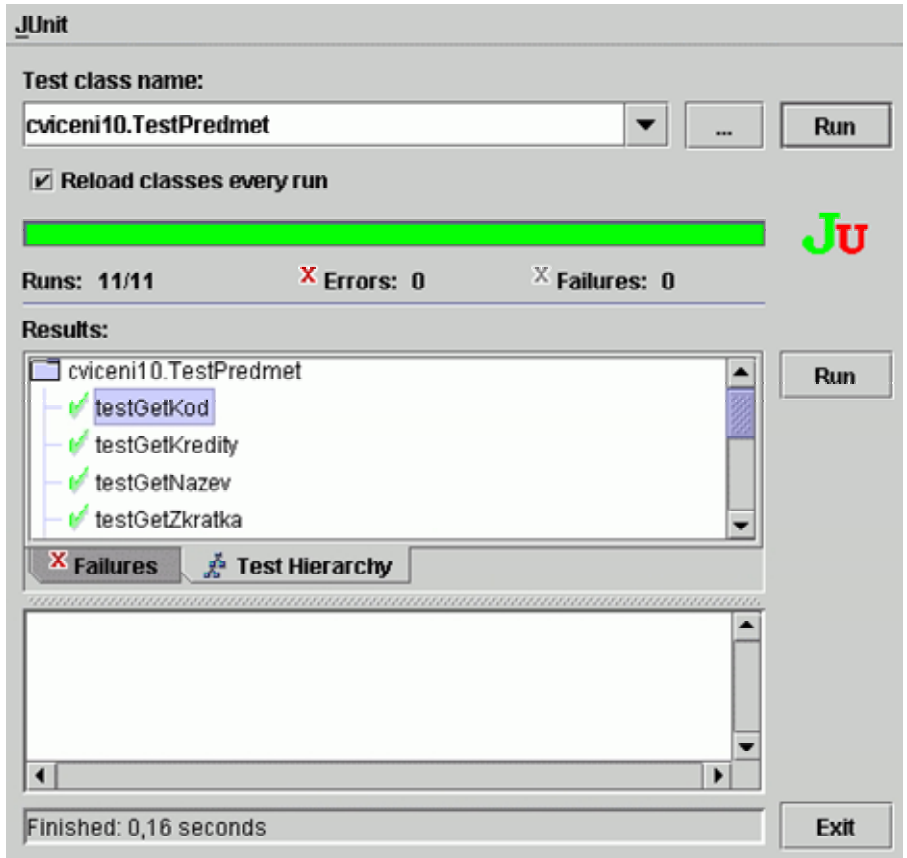
Přímé čtení záznamu

```
int getRecordCount(RandomAccessFile file)
{
  return (int)(file.length() / DELKA_ZAZNAMU);
}

Predmet read(RandomAccessFile file,
             int index)
{
  int offset = index * DELKA_ZAZNAMU;
  file.seek(offset);
  return read(file);
}
```

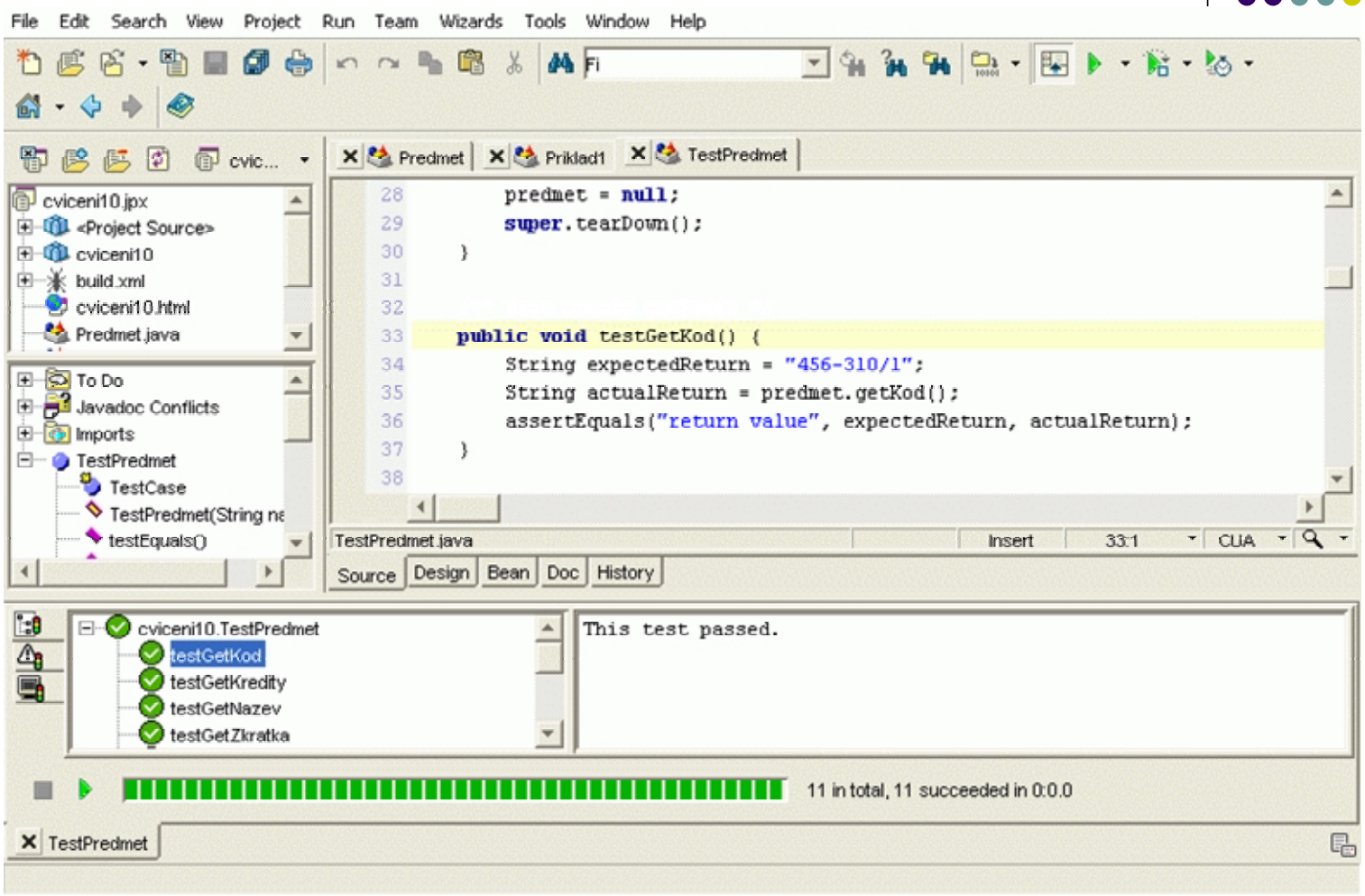
Přímý zápis záznamu

```
void write(RandomAccessFile file, int index)
    throws IOException
{
    // nastavíme pozici v souboru
    file.seek(index * DELKA_ZAZNAMU);

    // uložíme záznam
    write(file);
}
```

Systematické testování



16


File Edit Search View Project Run Team Wizards Tools Window Help
cviceni10.jpx
<Project Source>
cviceni10
build.xml
cviceni10.html
Predmet.java
To Do
Javadoc Conflicts
Imports
TestPredmet
TestCase
testEquals()
Predmet
Priklad1
TestPredmet
28 predmet = null;
29 super.tearDown();
30 }
33 public void testGetKod() {
34 String expectedReturn = "456-310/1";
35 String actualReturn = predmet.getKod();
36 assertEquals("return value", expectedReturn, actualReturn);
37 }
TestPredmet.java
Insert
33:1
CUA
Source
Design
Bean
Doc
History
cviceni10.TestPredmet
testGetKod
testGetKredity
testGetNazev
testGetZkratka
This test passed.
11 in total, 11 succeeded in 0.0.0
TestPredmet

Systematické testování

```
public class TestPredmet extends TestCase {
  private Predmet predmet = null;

  protected void setUp() throws Exception {
    super.setUp();
    predmet = new Predmet("456-310/1", "FLP",
       "Funkcionální a logické programování", 6);
  }

  protected void tearDown() throws Exception {
    predmet = null;
    super.tearDown();
  }
```

Systematické testování

```
public void testGetKod()
{
   String ocekavany = "456-310/1";
   String skutecny  = predmet.getKod();
   assertEquals("kod", ocekavany, skutecny);
}

public void testSetKod() {
   String kod = "123-456/7";
   predmet.setKod(kod);
   assertEquals(“kod”, kod, predmet.getKod());
}
```

Systematické testování

```
protected void setUp() throws Exception
{
  super.setUp();
  file = new RandomAccessFile("test.bin",
                              "rw");
  file.setLength(0); // smazání obsahu
  predmet1 = new Predmet("456-310/1",
                         "FLP1", "Predmet 1", 3);
  predmet2 = new Predmet("456-310/2",
                         "FLP2", "Predmet 2", 6);
}
```

Systematické testování

```
public void testRead() throws IOException {
  predmet1.write(file);
  predmet2.write(file);
  file.seek(0);

  Predmet pred = Predmet.read(file);
  assertNotNull("cteni prvniho zaznamu", pred);
  assertEquals("prvni zaznam", predmet1, pred);

  pred = Predmet.read(file);
  assertNotNull("cteni druheho zaznamu", pred);
  assertEquals("druhy zaznam", predmet2, pred);

  pred = Predmet.read(file);
  assertNull("cteni za koncem vrati null", pred);
}
```

Systematické testování

```
void testWrite() throws IOException {
  predmet1.write(file);
  file.seek(0);
  Predmet pred = Predmet.read(file);

  assertEquals("kod", predmet1.getKod(),
     pred.getKod());
  assertEquals("zkratka", predmet1.getZkratka(),
     pred.getZkratka());
  assertEquals("nazev", predmet1.getNazev(),
     pred.getNazev());
  assertEquals("kredity", predmet1.getKredity(),
     pred.getKredity());
}
```

Úkol do cvičení

1. Prostudovat ukázkový příklad na práci se souborem typu RandomAccessFile
2. Prostudovat testy k ukázkovému příkladu.
3. Vytvořit vlastní testy na ověření třídy Zlomek
   - ověření zadaných podmínek
   - nalezení a ověření dalších podmínek
   - použití nástrojů pro testování (JBuilder, Ant, grafické rozhraní JUnit)