A0B36PR2 – test Java 11.4.2013

Hodnocení (max. 10 bodů)

Jméno: _____ Login: _____ Cvičení: _____

Maximální počet bodů z testu je 10. Správná odpověď na testové otázky je taková, kde jsou označeny správně všechny možnosti (může jich být více). Správná odpověď je hodnocena 1 bodem.
Instrukce: označená odpověď ⊠, oprava (odznačení) ■. Oprava opravy není možná!

1. [max 1 bod] Výjimky - blok *finally* se provede...
   - □ vždy
   - □ pouze při provedení alespoň jedné klausule *catch*
   - □ pouze při provedení právě jedné klausule *catch*
   - □ pouze pokud není provedena žádná klausule *catch*

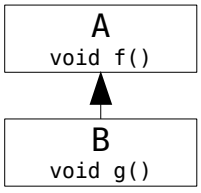
2. [max 1 bod] Máte okno s tlačítkem. Obsluha události stisku tlačítka může být realizována pomocí:
   - □ samostatné třídy deklarované jako *public* (tzv. handler)
   - □ samostatné třídy (handleru) deklarované v souboru popisujícím okno
   - □ vnitřní třídy
   - □ anonymní vnitřní třídy
   - □ implementace rozhraní *ActionListener* přímo třídou reprezentující okno

3. [max 1 bod] Máte následující část kódu:

```
int a;
int b = 10;
try {
   a = b / 0;
}
catch (RuntimeException e) {
   System.out.println("1");
}
catch (Exception e) {
   System.out.println("2");
}
```

   Výstupem tohoto programu na obrazovku bude:
   - □ "1"
   - □ "2"
   - □ "1" a "2"
   - □ *program nevypíše nic*

4. [max 1 bod] V jazyce Java definujte třídy *A* a *B* podle následující struktury (těla metod mohou být prázdná):



5. [max 1 bod] Pro třídy z předchozího příkladu a program `A a = new B();`
   Označte všechny správné možnosti volání:
   - □ `a.f()`
   - □ `a.g()`
   - □ `((B)a).g()`
   - □ `((B)a).toString()`

6. [max 2 body] Předpokládejte následující fragment kódu na serveru:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
BufferedReader systemIn = new BufferedReader(new InputStreamReader(System.in));
```

kde `socket` reprezentuje otevřené síťové spojení (instance třídy `Socket`).
Napište kód, který protistraně pošle jako řetězec náhodné reálné číslo v intervalu <0;1) formátované na 4 desetinná místa.

7. Implementujte jednoduchou třídu Student, která uchovává jméno, příjmení a studijní průměr studenta a metodu CompareTo pro řazení studentů podle studijního průměru [1 bod].
Uchování dat o jednotlivých studentech pomocí kolekcí (`ArrayList`) a řazení studentů demonstrujte na jednoduchém příkladu (vytvoření struktury, naplnění několika studenty, seřazení struktury) [1 bod].

8. [max 1 bod] Stručně vysvětlete klíčové slovo *synchronized.* Jakého problému se týká a jak ho řeší?

A0B36PR2 – test Java 11.4.2013

Hodnocení (max. 10 bodů)

Jméno: _____ Login: _____ Cvičení: _____

Maximální počet bodů z testu je 10. Správná odpověď na testové otázky je taková, kde jsou označeny správně všechny možnosti (může jich být více). Správná odpověď je hodnocena 1 bodem.
Instrukce: označená odpověď ⊠, oprava (odznačení) ■. Oprava opravy není možná!

1. [max 1 bod] Výjimky - blok *finally* se provede...
   - □ vždy
   - □ pouze při provedení alespoň jedné klausule *catch*
   - □ pouze při provedení právě jedné klausule *catch*
   - □ pouze pokud není provedena žádná klausule *catch*

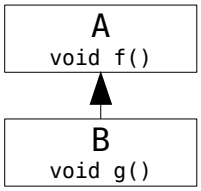
2. [max 1 bod] Máte okno s tlačítkem. Obsluha události stisku tlačítka může být realizována pomocí:
   - □ samostatné třídy deklarované jako *public* (tzv. handler)
   - □ samostatné třídy (handleru) deklarované v souboru popisujícím okno
   - □ vnitřní třídy
   - □ anonymní vnitřní třídy
   - □ implementace rozhraní *ActionListener* přímo třídou reprezentující okno

3. [max 1 bod] Máte následující část kódu:

```
int a;
int b = 10;
try {
   a = b / 0;
}
catch (RuntimeException e) {
   System.out.println("1");
}
catch (Exception e) {
   System.out.println("2");
}
```

   Výstupem tohoto programu na obrazovku bude:
   - □ "2"
   - □ "1"
   - □ "1" a "2"
   - □ *program nevypíše nic*

4. [max 1 bod] V jazyce Java definujte třídy *A* a *B* podle následující struktury (těla metod mohou být prázdná):



5. [max 1 bod] Pro třídy z předchozího příkladu a program `B a = new B();`
   Označte všechny správné možnosti volání:
   - □ `a.f()`
   - □ `a.g()`
   - □ `((B)a).g()`
   - □ `((B)a).toString()`

6. [max 2 body] Předpokládejte následující fragment kódu na serveru:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
BufferedReader systemIn = new BufferedReader(new InputStreamReader(System.in));
```

kde `socket` reprezentuje otevřené síťové spojení (instance třídy `Socket`).
Napište kód, který protistraně pošle jako řetězec náhodné reálné číslo v intervalu <0;1) formátované na 4 desetinná místa.

7. Implementujte jednoduchou třídu Student, která uchovává jméno, příjmení a studijní průměr studenta a metodu CompareTo pro řazení studentů podle studijního průměru [1 bod].
Uchování dat o jednotlivých studentech pomocí kolekcí (`ArrayList`) a řazení studentů demonstrujte na jednoduchém příkladu (vytvoření struktury, naplnění několika studenty, seřazení struktury) [1 bod].

8. [max 1 bod] Stručně vysvětlete klíčové slovo *synchronized.* Jakého problému se týká a jak ho řeší?

A0B36PR2 – test Java 11.4.2013

Hodnocení (max. 10 bodů)

Jméno: _____ Login: _____ Cvičení: _____

Maximální počet bodů z testu je 10. Správná odpověď na testové otázky je taková, kde jsou označeny správně všechny možnosti (může jich být více). Správná odpověď je hodnocena 1 bodem.
Instrukce: označená odpověď ⊠, oprava (odznačení) ■. Oprava opravy není možná!

1. [max 1 bod] Výjimky - blok *finally* se provede...
   - □ nikdy
   - □ pouze při provedení alespoň jedné klausule *catch*
   - □ vždy
   - □ až po skončení metody *main*

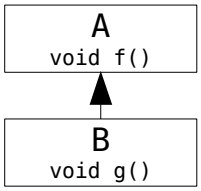
2. [max 1 bod] Máte okno s tlačítkem. Obsluha události stisku tlačítka může být realizována pomocí:
   - □ samostatné třídy deklarované jako *public* (tzv. handler)
   - □ samostatné třídy (handleru) deklarované v souboru popisujícím okno
   - □ vnitřní třídy
   - □ anonymní vnitřní třídy
   - □ implementace rozhraní *ActionListener* přímo třídou reprezentující okno

3. [max 1 bod] Máte následující část kódu:

```
int a;
int b = 10;
try {
   a = b / 0;
}
catch (Exception e) {
   System.out.println("Error");
}
catch (Error e) {
   System.out.println("Exception");
}
```

   Výstupem tohoto programu na obrazovku bude:
   - □ "Error"
   - □ "Exception"
   - □ "Error" a "Exception"
   - □ *program nevypíše nic*

4. [max 1 bod] V jazyce Java definujte třídy *A* a *B* podle následující struktury (těla metod mohou být prázdná):



5. [max 1 bod] Pro třídy z předchozího příkladu a program `A a = new B();`
   Označte všechny správné možnosti volání:
   - □ `a.g()`
   - □ `a.f()`
   - □ `((B)a).g()`
   - □ *taková inicializace nedává smysl a kód nepůjde přeložit*

6. [max 2 body] Předpokládejte následující fragment kódu na serveru:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
BufferedReader systemIn = new BufferedReader(new InputStreamReader(System.in));
```

kde `socket` reprezentuje otevřené síťové spojení (instance třídy `Socket`).
Napište kód, který čte data od klienta, tiskne je na obrazovku a program ukončí v případě, že klient pošle řetězec "Bye".

7. Implementujte jednoduchou třídu `Student`, která uchovává jméno, příjmení a studijní průměr studenta a třídu `StudentComparator` implementující komparátor umožňující řazení studentů podle studijního průměru [1 bod].
Uchování dat o jednotlivých studentech pomocí kolekcí (`ArrayList`) a řazení studentů demonstrujte na jednoduchém příkladu (vytvoření struktury, naplnění několika studenty, seřazení struktury) [1 bod].

8. [max 1 bod] Upravte kód následující třídy tak, aby byl bezpečně použitelný ve vícevláknovém programu.

```
public class EvenGen {
	int i = 0;
	void next() {
		i++;
		i++;
	}

	int getVal() {
		return i;
	}
}
```

A0B36PR2 – test Java 11.4.2013

Hodnocení (max. 10 bodů)

Jméno: _____ Login: _____ Cvičení: _____

Maximální počet bodů z testu je 10. Správná odpověď na testové otázky je taková, kde jsou označeny správně všechny možnosti (může jich být více). Správná odpověď je hodnocena 1 bodem.
Instrukce: označená odpověď ⊠, oprava (odznačení) ■. Oprava opravy není možná!

1. [max 1 bod] Výjimky - blok *finally* se provede...
   - □ nikdy
   - □ pouze při provedení alespoň jedné klausule *catch*
   - □ vždy
   - □ až po skončení metody *main*

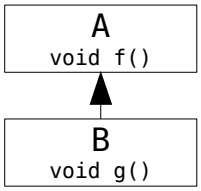
2. [max 1 bod] Máte okno s tlačítkem. Obsluha události stisku tlačítka může být realizována pomocí:
   - □ samostatné třídy deklarované jako *public* (tzv. handler)
   - □ samostatné třídy (handleru) deklarované v souboru popisujícím okno
   - □ vnitřní třídy
   - □ anonymní vnitřní třídy
   - □ implementace rozhraní *ActionListener* přímo třídou reprezentující okno

3. [max 1 bod] Máte následující část kódu:

```
int a;
int b = 10;
try {
   a = b / 0;
}
catch (Exception e) {
   System.out.println("Error");
}
catch (Error e) {
   System.out.println("Exception");
}
```

   Výstupem tohoto programu na obrazovku bude:
   - □ "Error"
   - □ "Exception"
   - □ "Error" a "Exception"
   - □ *program nevypíše nic*

4. [max 1 bod] V jazyce Java definujte třídy *A* a *B* podle následující struktury (těla metod mohou být prázdná):



5. [max 1 bod] Pro třídy z předchozího příkladu a program `A a = new B();`
   Označte všechny správné možnosti volání:
   - □ `a.g()`
   - □ `a.f()`
   - □ `((B)a).g()`
   - □ *taková inicializace nedává smysl a kód nepůjde přeložit*

6. [max 2 body] Předpokládejte následující fragment kódu na serveru:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
BufferedReader systemIn = new BufferedReader(new InputStreamReader(System.in));
```

kde `socket` reprezentuje otevřené síťové spojení (instance třídy `Socket`).
Napište kód, který čte data od klienta, tiskne je na obrazovku a program ukončí v případě, že klient pošle řetězec “Bye”.

7. Implementujte jednoduchou třídu `Student`, která uchovává jméno, příjmení a studijní průměr studenta a třídu `StudentComparator` implementující komparátor umožňující řazení studentů podle studijního průměru [1 bod].
Uchování dat o jednotlivých studentech pomocí kolekcí (`ArrayList`) a řazení studentů demonstrujte na jednoduchém příkladu (vytvoření struktury, naplnění několika studenty, seřazení struktury) [1 bod].

8. [max 1 bod] Proveďte nezbytně nutné změny v následujícícm kódu tak, aby byl bezpečně použitelný ve vícevláknovém programu.

```
public class Gen {
	int i = 0;
	void next() {
		i++;
	}

	int getVal() {
		return i;
	}
}
```

A0B36PR2 – test Java 11.4.2013

Hodnocení (max. 10 bodů)

Jméno: _____ Login: _____ Cvičení: _____

Maximální počet bodů z testu je 10. Správná odpověď na testové otázky je taková, kde jsou označeny správně všechny možnosti (může jich být více). Správná odpověď je hodnocena 1 bodem.
Instrukce: označená odpověď ⊠, oprava (odznačení) ■. Oprava opravy není možná!

1. [max 1 bod] Výjimky - blok *finally* se provede...
   - □ pouze pokud v bloku *try* nenastane výjimka
   - □ pouze při provedení všech klausulí *catch*
   - □ vždy, bez ohledu na to, zda v bloku *try* došlo nebo nedošlo k výjimce
   - □ pouze při provedení alespoň jedné klausule *catch*

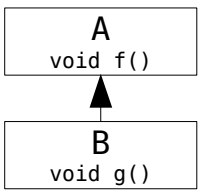
2. [max 2 body] Stručně popište pojmy kontejner a komponenta (v souvislosti s GUI v Javě) a uveďte u každého pojmu alespoň dva příklady (stačí název třídy).

3. [max 1 bod] Máte následující část kódu:

```
int a;
int b = 10;
try {
   a = b / 0;
}
catch (Error e) {
   System.out.println("Exception");
}
catch (Exception e) {
   System.out.println("Error");
}
```

   Výstupem tohoto programu na obrazovku bude:
   - □ "Error"
   - □ "Exception"
   - □ "Error" a "Exception"
   - □ *program nevypíše nic*

4. [max 1 bod] V jazyce Java definujte třídy *A* a *B* podle následující struktury (těla metod mohou být prázdná):



5. [max 1 bod] Pro třídy z předchozího příkladu a program `B a = new B();`
   Označte všechny správné možnosti volání:
   - □ `a.g()`
   - □ `a.f()`
   - □ `((B)a).g()`
   - □ *taková inicializace nedává smysl a kód nepůjde přeložit*

6. [max 2 body] Předpokládejte následující fragment kódu na serveru:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
BufferedReader systemIn = new BufferedReader(new InputStreamReader(System.in));
```

kde `socket` reprezentuje otevřené síťové spojení (instance třídy `Socket`).
Napište kód realizující tzv. echo server – klientovi pošle stejný text, který od něj obdržel.

7. Implementujte jednoduchou třídu `Complex`, která uchovává hodnotu komplexního čísla v algebraickém tvaru a a třídu `ComplexAbsComparator` implementující komparátor umožňující řazení těchto čísel podle absolutní hodnoty [1 bod].

8. [max 1 bod] Upravte kód následující třídy tak, aby byl bezpečně použitelný ve vícevláknovém programu.

```
public class EvenGen {
	int i = 0;
	void next() {
		i++;
		i++;
	}

	int getVal() {
		return i;
	}
}
```

A0B36PR2 – test Java 11.4.2013

Hodnocení (max. 10 bodů)

Jméno: _____ Login: _____ Cvičení: _____

Maximální počet bodů z testu je 10. Správná odpověď na testové otázky je taková, kde jsou označeny správně všechny možnosti (může jich být více). Správná odpověď je hodnocena 1 bodem.
Instrukce: označená odpověď ⊠, oprava (odznačení) ■. Oprava opravy není možná!

1. [max 1 bod] Výjimky - blok *finally* se provede...
   - □ pouze pokud v bloku *try* nenastane výjimka
   - □ pouze při provedení všech klausulí *catch*
   - □ vždy, bez ohledu na to, zda v bloku *try* došlo nebo nedošlo k výjimce
   - □ pouze při provedení alespoň jedné klausule *catch*

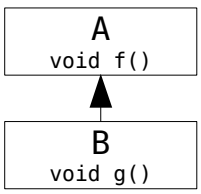
2. [max 2 body] Stručně popište pojmy kontejner a komponenta (v souvislosti s GUI v Javě) a uveďte u každého pojmu alespoň dva příklady (stačí název třídy).

3. [max 1 bod] Máte následující část kódu:

```
int a;
int b = 10;
try {
   a = b / 0;
}
catch (Error e) {
   System.out.println("Exception");
}
catch (Exception e) {
   System.out.println("Error");
}
```

   Výstupem tohoto programu na obrazovku bude:
   - □ "Error"
   - □ "Exception"
   - □ "Error" a "Exception"
   - □ *program nevypíše nic*

4. [max 1 bod] V jazyce Java definujte třídy *A* a *B* podle následující struktury (těla metod mohou být prázdná):



5. [max 1 bod] Pro třídy z předchozího příkladu a program `B a = new B();`
   Označte všechny správné možnosti volání:
   - □ `a.g()`
   - □ `a.f()`
   - □ `((B)a).g()`
   - □ *taková inicializace nedává smysl a kód nepůjde přeložit*

6. [max 2 body] Předpokládejte následující fragment kódu na serveru:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
BufferedReader systemIn = new BufferedReader(new InputStreamReader(System.in));
```

kde `socket` reprezentuje otevřené síťové spojení (instance třídy `Socket`).
Napište kód realizující tzv. echo server – klientovi pošle stejný text, který od něj obdržel.

7. Implementujte jednoduchou třídu `Complex`, která uchovává hodnotu komplexního čísla v algebraickém tvaru a metodu `CompareTo` implementující komparátor umožňující řazení těchto čísel podle absolutní hodnoty [1 bod].

8. [max 1 bod] Upravte kód následující třídy tak, aby byl bezpečně použitelný ve vícevláknovém programu.

```
public class EvenGen {
	int i = 0;
	void next() {
		i++;
		i++;
	}

	int getVal() {
		return i;
	}
}
```

A0B36PR2 – test Java 11.4.2013

Hodnocení (max. 10 bodů)

Jméno: _____ Login: _____ Cvičení: _____

Maximální počet bodů z testu je 10. Správná odpověď na testové otázky je taková, kde jsou označeny správně všechny možnosti (může jich být více). Správná odpověď je hodnocena 1 bodem.
Instrukce: označená odpověď ⊠, oprava (odznačení) ■. Oprava opravy není možná!

1. [max 1 bod] Blok *synchronized* ...
   - □ může být prováděn v jednu chvíli právě jedním vláknem
   - □ může být prováděn v jednu chvíli několika vlákny
   - □ se provede pouze na počítači s více procesory (procesorovými jádry)
   - □ slouží k synchronizaci programu s hodinami (taktovací frekvencí) procesoru

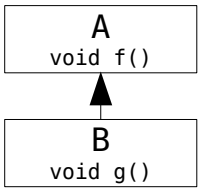
2. [max 1 bod] Máte okno s tlačítkem. Obsluha události stisku tlačítka může být realizována pomocí:
   - □ samostatné třídy deklarované jako *public* (tzv. handler)
   - □ samostatné třídy (handleru) deklarované v souboru popisujícím okno
   - □ vnitřní třídy
   - □ anonymní vnitřní třídy
   - □ implementace rozhraní *ActionListener* přímo třídou reprezentující okno

3. [max 1 bod] Máte následující část kódu:

```
int a;
int b = 10;
try {
   a = b / 0;
}
catch (RuntimeException e) {
   System.out.println("RuntimeException");
}
catch (Exception e) {
   System.out.println("Exception");
}
```

   Výstupem tohoto programu na obrazovku bude:
   - □ "RuntimeException"
   - □ "Exception"
   - □ "RuntimeException" a "Exception"
   - □ *takový program nelze přeložit*
   - □ *program nevypíše nic*

4. [max 1 bod] V jazyce Java definujte třídy *A* a *B* podle následující struktury (těla metod mohou být prázdná):



5. [max 1 bod] Pro třídy z předchozího příkladu a program `B a = new B();`
   Označte všechny správné možnosti volání:
   - □ `a.f()`
   - □ `a.g()`
   - □ `((B)a).g()`
   - □ `((B)a).toString()`

6. [max 3 body] Předpokládejte následující fragment kódu na serveru:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
BufferedReader systemIn = new BufferedReader(new InputStreamReader(System.in));
```

   kde `socket` reprezentuje otevřené síťové spojení (instance třídy `Socket`).
   Napište kód, který přijímá po síťovém spojení celá čísla (posílaná jako textové řetězce) a protistraně posílá vždy součet posledních dvou.

7. Implementujte jednoduchou třídu `BankAccount`, která uchovává jméno majitele účtu, výši zůstatku na účtu (double) a třídu `AccountComparator` implementující komparátor umožňující řazení účtů podle zůstatku [1 bod].
   Uchování dat o jednotlivých účtech pomocí kolekcí (`ArrayList`) a řazení záznamů demonstrujte na jednoduchém příkladu (vytvoření struktury, naplnění několika záznamy, seřazení struktury) [1 bod].

A0B36PR2 – test Java 11.4.2013

Hodnocení (max. 10 bodů)

Jméno: _____ Login: _____ Cvičení: _____

Maximální počet bodů z testu je 10. Správná odpověď na testové otázky je taková, kde jsou označeny správně všechny možnosti (může jich být více). Správná odpověď je hodnocena 1 bodem.
Instrukce: označená odpověď ⊠, oprava (odznačení) ■. Oprava opravy není možná!

1. [max 1 bod] Blok *synchronized* ...
   - □ může být prováděn v jednu chvíli právě jedním vláknem
   - □ může být prováděn v jednu chvíli několika vlákny
   - □ se provede pouze na počítači s více procesory (procesorovými jádry)
   - □ slouží k synchronizaci programu s hodinami (taktovací frekvencí) procesoru

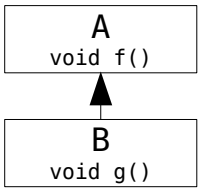
2. [max 1 bod] Máte okno s tlačítkem. Obsluha události stisku tlačítka může být realizována pomocí:
   - □ samostatné třídy deklarované jako *public* (tzv. handler)
   - □ samostatné třídy (handleru) deklarované v souboru popisujícím okno
   - □ vnitřní třídy
   - □ anonymní vnitřní třídy
   - □ implementace rozhraní *ActionListener* přímo třídou reprezentující okno

3. [max 1 bod] Máte následující část kódu:

```
int a;
int b = 10;
try {
   a = b / 0;
}
catch (RuntimeException e) {
   System.out.println("RuntimeException");
}
catch (Exception e) {
   System.out.println("Exception");
}
```

   Výstupem tohoto programu na obrazovku bude:
   - □ "RuntimeException"
   - □ "Exception"
   - □ "RuntimeException" a "Exception"
   - □ *takový program nelze přeložit*
   - □ *program nevypíše nic*

4. [max 1 bod] V jazyce Java definujte třídy *A* a *B* podle následující struktury (těla metod mohou být prázdná):



5. [max 1 bod] Pro třídy z předchozího příkladu a program `B a = new B();`
   Označte všechny správné možnosti volání:
   - □ `a.f()`
   - □ `a.g()`
   - □ `((B)a).g()`
   - □ `((B)a).toString()`

6. [max 3 body] Předpokládejte následující fragment kódu na serveru:

```
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
BufferedReader systemIn = new BufferedReader(new InputStreamReader(System.in));
```

   kde `socket` reprezentuje otevřené síťové spojení (instance třídy `Socket`).
   Napište kód, který přijímá po síťovém spojení celá čísla (posílaná jako textové řetězce) a protistraně posílá vždy součet posledních dvou.

7. Implementujte jednoduchou třídu `BankAccount`, která uchovává jméno majitele účtu, výši zůstatku na účtu (double) a metodu `CompareTo` pro řazení účtů podle zůstatku [1 bod].
   Uchování dat o jednotlivých účtech pomocí kolekcí (`ArrayList`) a řazení záznamů demonstrujte na jednoduchém příkladu (vytvoření struktury, naplnění několika záznamy, seřazení struktury) [1 bod].