**Semestrální práce – Obchod s HW**

**Tomáš Stark 1IT**

**Popis aplikace**

Aplikace, kterou jsem vyvinul, slouží k zjištění, zda je obchod otevřený či zavřený v závislosti na aktuálním čase a otevírací době obchodu. Při spuštění aplikace se na obrazovce ihned zobrazí informace o provozní době obchodu, a to, zda je momentálně otevřený nebo zavřený. Tato funkce je velmi užitečná pro zákazníky, kteří chtějí vědět, zda mají v obchodě ještě šanci nakoupit, nebo zda je nutné počkat do příštího pracovního dne.

Ale to není vše, co tato aplikace nabízí. Po zobrazení stavu obchodu se na obrazovce objeví aktuální nabídka laptopů a desktopů, spolu s jejich cenami a přirážkou k ceně. Uživatelé mohou jednoduše procházet dostupné modely a jejich parametry, aby si vybrali zařízení, které nejlépe vyhovuje jejich potřebám. Kromě toho jsou všechny specifikace zařízení přehledně zobrazeny, což uživatelům umožňuje rychle srovnávat různé možnosti.

Aplikace je navržena tak, aby poskytovala uživatelům přesné a aktuální informace. Je synchronizována s časem na zařízení, na kterém je spuštěna, a s databází, která obsahuje otevírací doby obchodu a nabídky zařízení. To znamená, že uživatelé mohou vždy spolehnout na to, že získají aktuální informace o otevřenosti obchodu a nabízených produktech.

Jednou z výhod této aplikace je také možnost přizpůsobit si parametry vyhledávání. Uživatelé mohou filtrovat nabídku laptopů a desktopů podle svých požadavků, jako je operační systém, procesor, paměť a úložný prostor. To zajišťuje, že uživatelé budou mít možnost vybrat si zařízení, které přesně splňuje jejich potřeby a preferované specifikace.

Celkově tato aplikace poskytuje uživatelům jednoduchý a efektivní způsob, jak zjistit, zda je obchod otevřený a jaké zařízení je momentálně k dispozici. S jejím uživatelsky přívětivým rozhraním a aktuálními informacemi je ideálním společníkem pro ty, kteří hledají nový laptop nebo desktop.

**Snímky kodu**

Třída počítač

package SemestralniPrace;  
  
*/\*\*  
 \* Třída reprezentující počítač.  
 \*  
 \* @author Tomas Stark  
 \* @ version 2023 - 05 - 14  
 \*/*public class Pocitac {  
  
 */\*\*  
 \* Značka počítače.  
 \*/* protected String znacka;  
  
 */\*\*  
 \* Cena počítače.  
 \*/* protected double cena;  
 protected String grafika;  
  
 */\*\*  
 \* Konstruktor pro vytvoření nového počítače s danou značkou a cenou.  
 \*  
 \* @param znacka Značka počítače.  
 \* @param cena Cena počítače.  
 \*/* public Pocitac(String *znacka*, double *cena*) {  
 this.znacka = *znacka*;  
 this.cena = *cena*;  
 }  
  
 */\*\*  
 \* Vrátí značku počítače.  
 \*  
 \* @return Značka počítače.  
 \*/* public String getZnacka() {  
 return znacka;  
 }  
  
 */\*\*  
 \* Nastaví značku počítače.  
 \*  
 \* @param znacka Značka počítače.  
 \*/* public void setZnacka(String *znacka*) {  
 this.znacka = *znacka*;  
 }  
  
 */\*\*  
 \* Vrátí cenu počítače.  
 \*  
 \* @return Cena počítače.  
 \*/* public double getCena() {  
 return cena;  
 }  
  
 */\*\*  
 \* Nastaví cenu počítače.  
 \*  
 \* @param cena Cena počítače.  
 \*/* public void setCena(double *cena*) {  
 this.cena = *cena*;  
 }  
}

Třída Laptop

package SemestralniPrace;  
  
*/\*\*  
 \* Třída reprezentující notebook.  
 \*  
 \* @author Tomas Stark  
 \* @ version 2023 - 05 - 14  
 \*/*public class Laptop extends Pocitac {  
  
 */\*\*  
 \* Velikost obrazovky notebooku v palcích.  
 \*/* private double VelikostObrazovky;  
  
 */\*\*  
 \* Velikost RAM paměti notebooku v GB.  
 \*/* private int ram;  
  
 */\*\*  
 \* Typ procesoru notebooku.  
 \*/* private String processor;  
  
 */\*\*  
 \* Konstruktor pro vytvoření nového notebooku s danou značkou, cenou, velikostí obrazovky, RAM a procesorem.  
 \*  
 \* @param znacka Značka notebooku.  
 \* @param cena Cena notebooku.  
 \* @param VelikostObrazovky Velikost obrazovky notebooku v palcích.  
 \* @param ram Velikost RAM paměti notebooku v GB.  
 \* @param processor Typ procesoru notebooku.  
 \*/* public Laptop(String *znacka*, double *cena*, double *VelikostObrazovky*, int *ram*, String *processor*) {  
 super(*znacka*, *cena*);  
 this.VelikostObrazovky = *VelikostObrazovky*;  
 this.ram = *ram*;  
 this.processor = *processor*;  
 }  
  
 */\*\*  
 \* Vrátí velikost obrazovky notebooku.  
 \*  
 \* @return Velikost obrazovky notebooku v palcích.  
 \*/* public double getVelikostObrazovky() {  
 return VelikostObrazovky;  
 }  
  
 */\*\*  
 \* Nastaví velikost obrazovky notebooku.  
 \*  
 \* @param velikostObrazovky Velikost obrazovky notebooku v palcích.  
 \*/* public void setVelikostObrazovky(double *velikostObrazovky*) {  
 this.VelikostObrazovky = *velikostObrazovky*;  
 }  
  
 */\*\*  
 \* Vrátí velikost RAM paměti notebooku.  
 \*  
 \* @return Velikost RAM paměti notebooku v GB.  
 \*/* public int getRam() {  
 return ram;  
 }  
  
 */\*\*  
 \* Nastaví velikost RAM paměti notebooku.  
 \*  
 \* @param ram Velikost RAM paměti notebooku v GB.  
 \*/* public void setRam(int *ram*) {  
 this.ram = *ram*;  
 }  
  
 */\*\*  
 \* Vrátí typ procesoru notebooku.  
 \*  
 \* @return Typ procesoru notebooku.  
 \*/* public String getProcessor() {  
 return processor;  
 }  
  
 */\*\*  
 \* Nastaví typ procesoru notebooku.  
 \*  
 \* @param processor Typ procesoru notebooku.  
 \*/* public void setProcessor(String *processor*) {  
 this.processor = *processor*;  
 }  
}

Třída Desktop

package SemestralniPrace;  
  
*/\*\*  
 \* Třída reprezentující stolní počítač.  
 \*  
 \* @author Tomas Stark  
 \* @ version 2023 - 05 - 14  
 \*/*public class Desktop extends Pocitac {  
 private double VelikostObrazovky;  
 private int ram;  
 private String processor;  
  
 */\*\*  
 \* Konstruktor třídy Desktop.  
 \*  
 \* @param znacka značka počítače  
 \* @param cena cena počítače  
 \* @param VelikostObrazovky velikost obrazovky v palcích  
 \* @param ram velikost operační paměti v GB  
 \* @param processor název procesoru  
 \* @param grafika název grafiky  
 \*/* public Desktop(String *znacka*, double *cena*, double *VelikostObrazovky*, int *ram*, String *processor*, String *grafika*) {  
 super(*znacka*, *cena*);  
 this.VelikostObrazovky = *VelikostObrazovky*;  
 this.ram = *ram*;  
 this.processor = *processor*;  
 this.grafika = *grafika*;  
 }  
  
 */\*\*  
 \* Vrátí velikost obrazovky počítače.  
 \*  
 \* @return velikost obrazovky v palcích  
 \*/* public double getVelikostObrazovky() {  
 return VelikostObrazovky;  
 }  
  
 */\*\*  
 \* Nastaví velikost obrazovky počítače.  
 \*  
 \* @param velikostObrazovky velikost obrazovky v palcích  
 \*/* public void setVelikostObrazovky(double *velikostObrazovky*) {  
 this.VelikostObrazovky = *velikostObrazovky*;  
 }  
  
 */\*\*  
 \* Vrátí velikost operační paměti počítače.  
 \*  
 \* @return velikost operační paměti v GB  
 \*/* public int getRam() {  
 return ram;  
 }  
  
 */\*\*  
 \* Nastaví velikost operační paměti počítače.  
 \*  
 \* @param ram velikost operační paměti v GB  
 \*/* public void setRam(int *ram*) {  
 this.ram = *ram*;  
 }  
  
 */\*\*  
 \* Vrátí název procesoru počítače.  
 \*  
 \* @return název procesoru  
 \*/* public String getProcessor() {  
 return processor;  
 }  
  
 */\*\*  
 \* Nastaví název procesoru počítače.  
 \*  
 \* @param processor název procesoru  
 \*/* public void setProcessor(String *processor*) {  
 this.processor = *processor*;  
 }  
  
 */\*\*  
 \* Vrátí název grafikcé karty počítače.  
 \*  
 \* @return název grafiky  
 \*/* public String getGrafika() {  
 return grafika;  
 }  
  
 public void setGrafika(String *grafika*) {  
 this.processor = *grafika*;  
 }  
}

Abstraktní třída Otevírací doba

package SemestralniPrace;  
  
*/\*\*  
 \* Abstraktní třída {@code OtevíracíDoba} reprezentuje otevírací dobu.  
 \* <p>  
 \* Tato třída slouží jako nadřazená třída pro specifické třídy reprezentující  
 \* otevírací doby různých podniků, jako jsou například obchody, restaurace, kavárny  
 \* a podobně.  
 \*  
 \* @author Tomas Stark  
 \* @ version 2023 - 05 - 14  
 \*/*public abstract class OteviraciDoba {  
 */\*\*  
 \* Určuje, zda je podnik v aktuální době otevřen.  
 \*  
 \* @return {@code true}, pokud je podnik otevřen, jinak {@code false}  
 \*/* public abstract boolean jeOtevreno();  
  
 */\*\*  
 \* Vrací textový řetězec, který popisuje, zda je podnik otevřen nebo zavřen.  
 \*  
 \* @return textový řetězec "Je otevřeno." nebo "Je zavřeno."  
 \*/* public String vypisOtevreni() {  
 if (jeOtevreno()) {  
 return "Je otevřeno. Muzete si prohlednou aktualni nabidku";  
 } else {  
 return "Je zavřeno. Prijde jindy. Oteviraci doba je od 9:00 - 17:00. Na aktualni nabidku se muzete podivat nize";  
 }  
 }  
  
 public abstract boolean Otevreno();  
}

Třída moje otevírací doba

package SemestralniPrace;  
  
import java.time.LocalTime;  
  
*/\*\*  
 \* Třída {@code MojeOteviraciDoba} reprezentuje otevírací dobu obchodu.  
 \* Otevírací doba je od 9:00 do 17:00.  
 \*  
 \* @author Tomas Stark  
 \* @ version 2023 - 05 - 14  
 \*/*public class MojeOteviraciDoba extends OteviraciDoba {  
 */\*\*  
 \* Určuje, zda je obchod v aktuální době otevřen.  
 \*  
 \* @return {@code true}, pokud je obchod otevřen, jinak {@code false}  
 \*/* @Override  
 public boolean jeOtevreno() {  
 LocalTime *aktualniCas* = LocalTime.*now*();  
 LocalTime *oteviraOd* = LocalTime.*of*(9, 0);  
 LocalTime *zaviraOd* = LocalTime.*of*(17, 0);  
 return *aktualniCas*.isAfter(*oteviraOd*) && *aktualniCas*.isBefore(*zaviraOd*);  
 }  
  
 */\*\*  
 \* @return jeOtvreno  
 \*/* public boolean Otevreno() {  
 return false;  
 }  
}

Třída Počítačový Obchod

package SemestralniPrace;  
  
*/\*\*  
 \* Třída reprezentující obchod s počítači.  
 \*  
 \* @author Tomas Stark  
 \* @ version 2023 - 05 - 14  
 \*/*public class PocitacovyObchod {  
  
 */\*\*  
 \* Metoda main spouští aplikaci obchodu s počítači.  
 \* Vytvoří instanci třídy Laptop a Desktop a vypíše informace o nich.  
 \*  
 \* @param args argumenty příkazové řádky (nevyužito)  
 \*/* public static void main(String[] *args*) {  
 Laptop *laptop* = new Laptop("Apple", 64999.99, 16.2, 16, "Apple M1 Pro");  
 Desktop *desktop* = new Desktop("Custom", 129999.99, 34.1, 32, "AMD Ryzen 9 5900x", "Nvidia RTX 4090");  
 MojeOteviraciDoba *oteviraciDoba* = new MojeOteviraciDoba();  
  
 final double *prirazka* = 21.0;  
 double *LaptopSprirazkou* = *laptop*.getCena() + (*laptop*.getCena() \* (*prirazka* / 100));  
 double *DesktopSprirazkou* = *desktop*.getCena() + (*desktop*.getCena() \* (*prirazka* / 100));  
  
 System.*out*.println();  
 System.*out*.println(*oteviraciDoba*.vypisOtevreni());  
 System.*out*.println();  
 System.*out*.println("Aktualni nabidka");  
 System.*out*.println("Laptop:");  
 System.*out*.println("Znacka: " + *laptop*.getZnacka());  
 System.*out*.println("Cena: " + *laptop*.getCena() + " Kč");  
 System.*out*.println("Cena s obchodni prirazkou: " + String.*format*("%.2f", *LaptopSprirazkou*) + " Kč");  
 System.*out*.println("Velikost obrazovky: " + *laptop*.getVelikostObrazovky());  
 System.*out*.println("RAM: " + *laptop*.getRam());  
 System.*out*.println("Procesor: " + *laptop*.getProcessor());  
  
 System.*out*.println();  
  
 System.*out*.println("Desktop:");  
 System.*out*.println("Znacka: " + *desktop*.getZnacka());  
 System.*out*.println("Cena: " + *desktop*.getCena() + " Kč");  
 System.*out*.println("Velikost obrazovky: " + *desktop*.getVelikostObrazovky());  
 System.*out*.println("Cena s obchodni prirazkou: " + String.*format*("%.2f", *DesktopSprirazkou*) + " Kč");  
 System.*out*.println("RAM: " + *desktop*.getRam());  
 System.*out*.println("Processor: " + *desktop*.getProcessor());  
 System.*out*.println("Grafická karta: " + *desktop*.getGrafika());  
  
 }  
}