

Произдственная практика
по специальности
09.02.07 Информационные системы и
программирование

Тема: «Разработка информационно-справочной системы
«Персональные компьютеры»»

Выполнил студент группы ИС-32 Суворов А.А.

Цель проекта - разработка информационно-справочной системы о Персональных компьютерах.

Объект исследования - информационно-справочные системы.

Предмет исследования - особенности реализации информационно-справочной системы, а также средства реализации программного продукта.

Для достижения поставленной цели в данном проекте были решены следующие задачи:

- определены темы, которые будут реализованы
- спроектированы структура и функциональность приложений;
- изучены и выбраны средства разработки информационно-справочных систем;
- разработана графическая составляющая приложения1;
- реализована структура и функциональность Программного продукта;

Выбор Инструментальных средств



Visual studio 2022

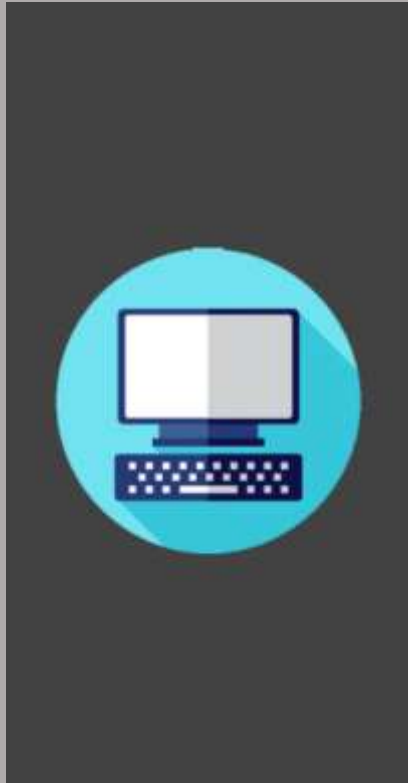


Android studio



Adobe photoshop

Мобильная версия



Экран загрузки

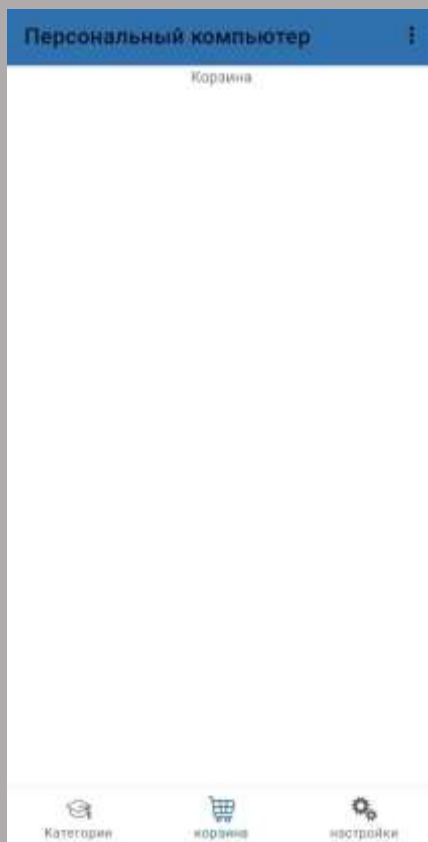


Главное меню



Настройки

Экран с корзиной

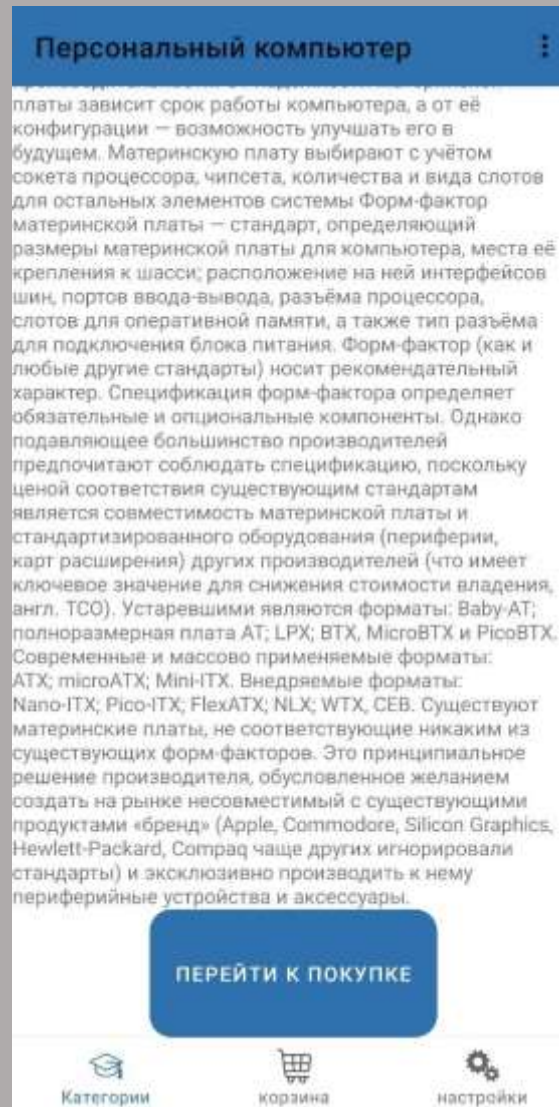


Пустая корзина

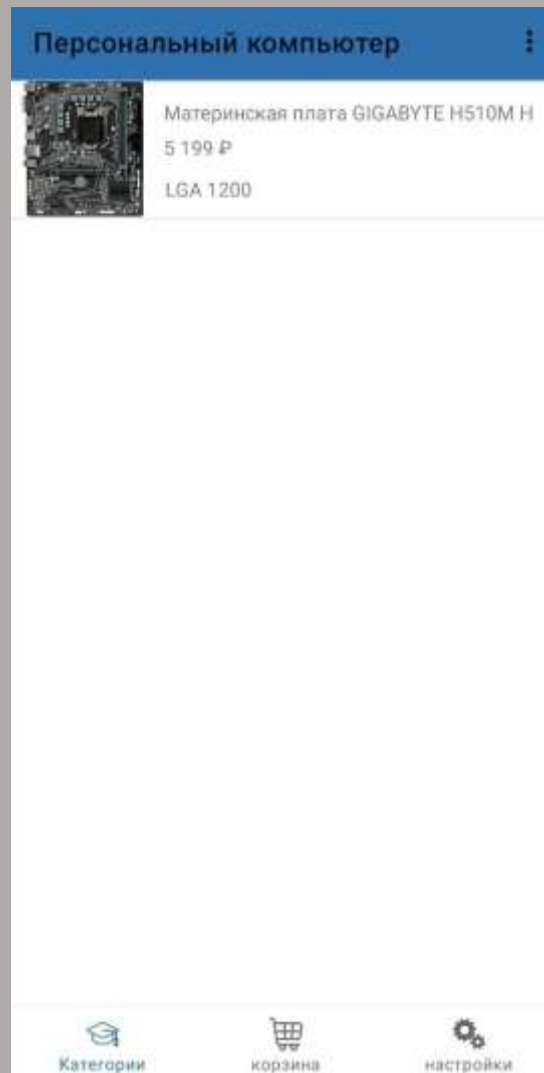


Корзина после покупки

Экран с выбранной лекцией



Экран покупки



Оптимизированная передача данных

```
class ViewModel1 : ViewModel() {  
    val name : MutableLiveData<String> by lazy {  
        MutableLiveData<String>()  
    }  
    val cena: MutableLiveData<String> by lazy {  
        MutableLiveData<String>()  
    }  
    val spec: MutableLiveData<String> by lazy {  
        MutableLiveData<String>()  
    }  
    val img: MutableLiveData<Int> by lazy {  
        MutableLiveData<Int>()  
    }  
}
```

Передача списка

```
override fun onCreateView(view: View, savedInstanceState: Bundle?) {  
    lateinit var adapter: CustomAdapter  
    val viewModels : ViewModel by activityViewModels()  
    val name : String = viewModels.name.value.toString()  
    val cena : String = viewModels.cena.value.toString()  
    val spec : String = viewModels.spec.value.toString()  
    val img : Int = viewModels.img.value!!.toInt()  
    val listView: ListView = view.findViewById<View>(R.id.spisok_pokupok) as ListView  
    val dataModels: ArrayList<dataModel> = ArrayList()  
    dataModels.add(  
        dataModel(  
            img,  
            name,  
            cena,  
            spec  
        )  
    )  
    adapter = CustomAdapter(dataModels, requireActivity().application)  
    listView.adapter = adapter  
    super.onCreateView(view, savedInstanceState)
```

Desktop-версия

Персональный компьютер

Лекции справка

 Материнская плата

 процессор

 Видеокарта

 Оперативная память

 жёсткий диск

 корпус

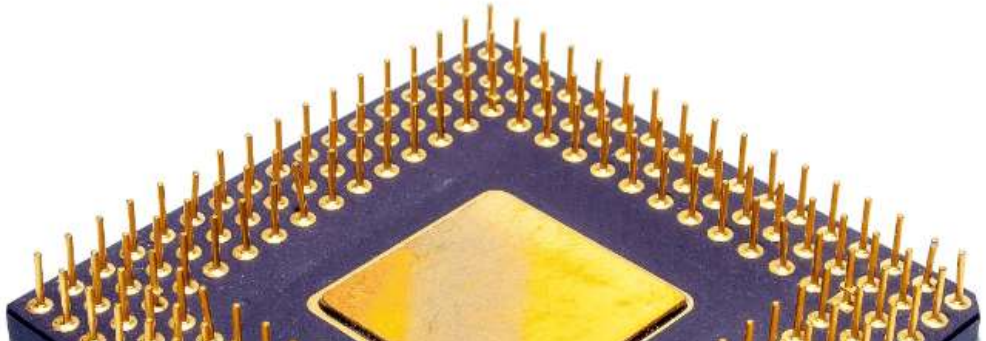
Что такое процессор (CPU)

В этой статье мы рассмотрим, что такое процессор CPU, какие у него функции и из чего он состоит.

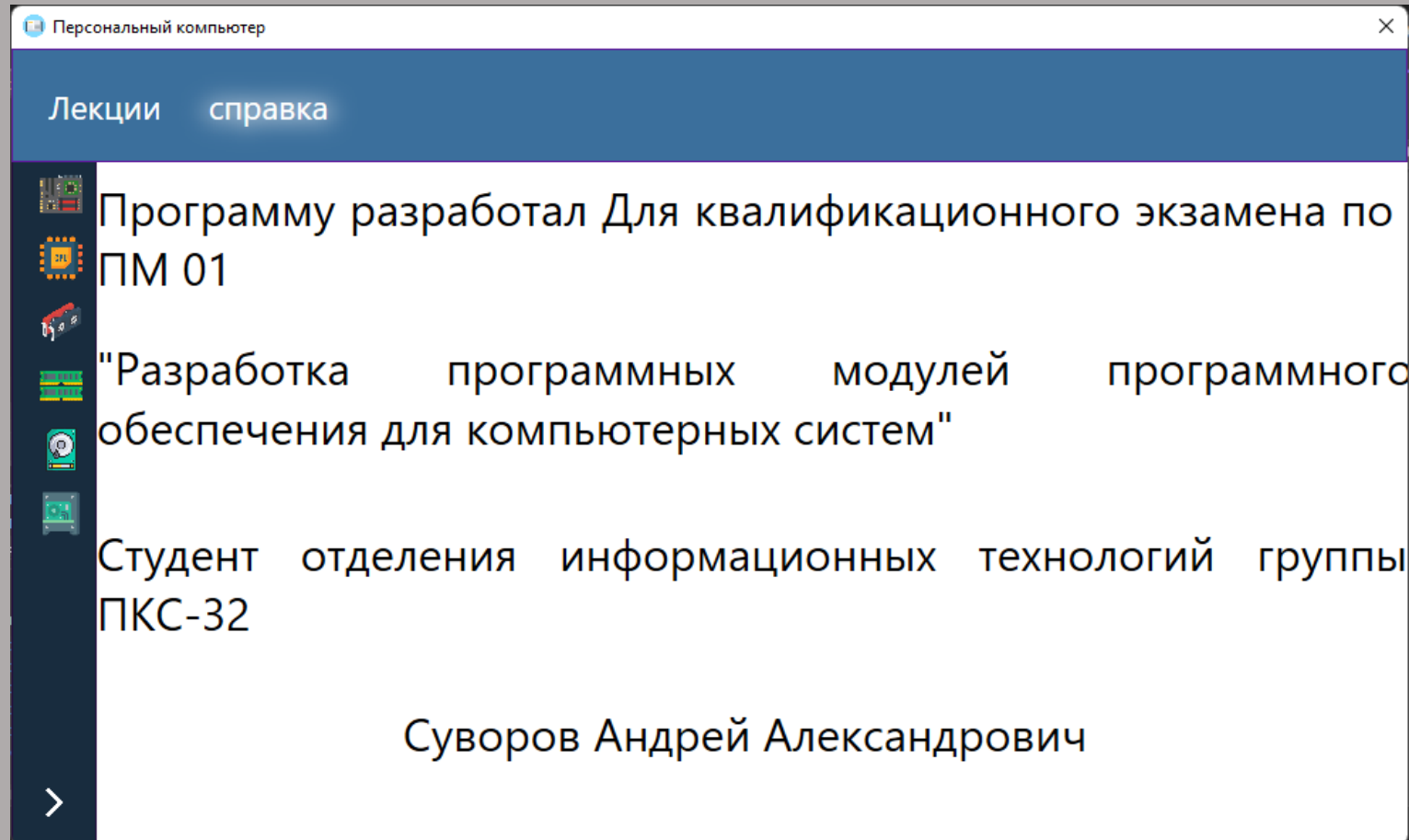
В каждом вычислительном устройстве (ПК, смартфон, фотоаппарат) есть центр, который отвечает за правильную работу машины — процессор.

В широком смысле процессор — это устройство, которое выполняет вычислительные и логические операции с данными. Чаще всего этот термин используется для обозначения центрального процессора устройства. Расшифровка CPU — Central Processing Unit (центральное обрабатывающее устройство). Это самая важная часть компьютера. Его мозг. Он выглядит как квадрат размером приблизительно 5х5 см:

С обратной стороны CPU находятся ножки, с помощью которых он крепится к материнской плате:



Окно справки



Взаимодействие с интерфейсом

```
<Label Content="{TemplateBinding Content}"
        Background="■"Transparent" FontSize="23"
        VerticalAlignment="Center">
  <Label.Style>
    <Style TargetType="Label">
      <Setter Property="Foreground" Value="■"White"/>
      <Style.Triggers>
        <DataTrigger Binding="{Binding Path=IsMouseOver, ElementName=Btn}"
                      Value="true">
          <Setter Property="Foreground" Value="■"White"/>
        </DataTrigger>
        <DataTrigger Binding="{Binding RelativeSource={RelativeSource
          Mode=FindAncestor,
          AncestorType={x:Type Button}},
          Path = IsPressed}" Value="true">
          <Setter Property="Foreground" Value="■"White"/>
        </DataTrigger>
        <DataTrigger Binding="{Binding RelativeSource={RelativeSource
          Mode=FindAncestor,
          AncestorType={x:Type Button}},
          Path = IsFocused}" Value="true">
          <Setter Property="Foreground" Value="■"White"/>
        </DataTrigger>
      </Style.Triggers>
    </Style>
  </Label.Style>
</Label>
```

Регистрация действий пользователя

```
namespace personal_computer.ViewModel
{
    Ссылка 1
    internal abstract class ViewModelBase : INotifyPropertyChanged
    {
        public event PropertyChangedEventHandler PropertyChanged;

        Ссылка 1
        protected virtual void OnPropertyChanged([CallerMemberName] string PropertyName = null)
        {
            PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(PropertyName));
        }

        Ссылка 1
        protected virtual bool Set<T>(ref T field, T value, [CallerMemberName] string PropertyName = null)
        {
            if (Equals(field, value)) return false;
            field = value;
            OnPropertyChanged(PropertyName);
            return true;
        }
    }
}
```